NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

OPTIMIZATION OF THE
UNITED STATES COAST GUARD
FORCE STRUCTURE

by

John E. Tomko

September 1991

Thesis Advisor: Siriphong Lawphongpanich

Approved for public release; distribution is unlimited

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL OR	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Including Security Classification)
OPTIMIZATION OF THE UNITED STATES COAST GUARD FORCE STRUCTURE

12 PERSONAL AUTHOR(S)
TOMKO, John E.

13 TYPE OF REPORT	13b. TIME COVERED FROM TO	14. DATE OF REPORT (Year, Month, Day)	15. Page Count
Master's thesis		1991, September	140

16. SUPPLEMENTAL NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Coast Guard, Optimization, Integer Programming, Force Structure, FORTRAN

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The United States Coast Guard area and district commands are assigned platforms to accomplish required missions. As the nature and profile of the missions change over time, the platforms must be reallocated to better satisfy the new mission needs. The problem of reallocating existing platforms and, perhaps, acquiring new ones is referred to as the force structure problem. This problem is complex because of the multi-mission capability of the platforms, and the difficulty in quantifying projected mission requirements and platform suitability in meeting these requirements.

In this thesis, methods for quantifying the mission requirements and platform suitability are proposed. Using these methods, a linear integer optimization model to allocate platforms is developed. Additionally, a FORTRAN based interface is implemented to facilitate the utilization of the model. Included in this interface are basic functions of a database system to aid the users in maintaining and updating model data. To demonstrate the use of the system, a realistic sample problem was extracted from Fiscal Year 1989 operational and administrative data. Outputs for the problem are given and analyzed.

20 DISTRIBUTION/AVAILABILTIY OF ABSTRACT	1a. REPORT SECURITY CLASSIFICATION	
[X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC	unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Siriphong Lawphongpanich	22b. TELEPHONE (Include Area Code) (408)646-2106	22c. OFFICE SYMBOL OR/Lp

DD Form 1473, JUN 86 — Previous editions are obsolete. S/N 0102-LF-014-6603 — SECURITY CLASSIFICATION OF THIS PAGE Unclassified

Approved for public release; distribution is unlimited.

Optimization of the
United States Coast Guard
Force Structure

by

John E. Tomko
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Author: _____
John E. Tomko

Approved by: _____
Siriphong Lawphongpanich, Thesis Advisor

_____
Gordon Bradley, Second Reader

_____
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

The United States Coast Guard area and district commands are assigned platforms to accomplish required missions. As the nature and profile of the missions change over time, the platforms must be reallocated to better satisfy the new mission needs. The problem of reallocating existing platforms and, perhaps, acquiring new ones is referred to as the force structure problem. This problem is complex because of the multi-mission capability of the platforms, and the difficulty in quantifying projected mission requirements and platform suitability in meeting these requirements.

In this thesis, methods for quantifying the mission requirements and platform suitability are proposed. Using these methods, a linear integer optimization model to allocate platforms is developed. Additionally, a FORTRAN based interface is implemented to facilitate the utilization of the model. Included in this interface are basic functions of a database system to aid the users in maintaining and updating model data. To demonstrate the use of the system, a realistic sample problem was extracted from Fiscal Year 1989 operational and administrative data. Outputs for the problem are given and analyzed.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logical errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION

The origin of the United States Coast Guard's motto **SEMPER PARATUS**, Always Ready, was lost in antiquity. The Coast Guard now faces many political, technological, and economic challenges that must be addressed to ensure that this motto remains an accurate description of the Coast Guard. The proper management of capital assets is necessary for the well-being of any organization. The Coast Guard's Force Structure problem can be broadly described as striving for the efficient procurement and operation of the Coast Guard's current and future fleet of cutters[1] and aircraft (or, more generally, platforms) to accomplish the myriad of mandated programs and missions.

A. BACKGROUND

The United States Coast Guard (USCG) is both operationally and administratively decentralized with its headquarters located in Washington DC. Two area commands, in New York and San Francisco, direct operations in the Atlantic and Pacific regions respectively. Area Commanders maintain operational control over cutters greater than 180 feet in length [Ref. 1].

---

[1]Cutters is the Coast Guard term for describing vessels sixty-five feet or longer.

The area commands are further divided into ten separate geographical commands called districts (see Figure 1). All aircraft and smaller cutters are allocated or assigned to districts. Each district is controlled by a Flag Officer who exercises autonomous authority over the platforms assigned to the district and uses them to accomplish missions within the district's boundaries.

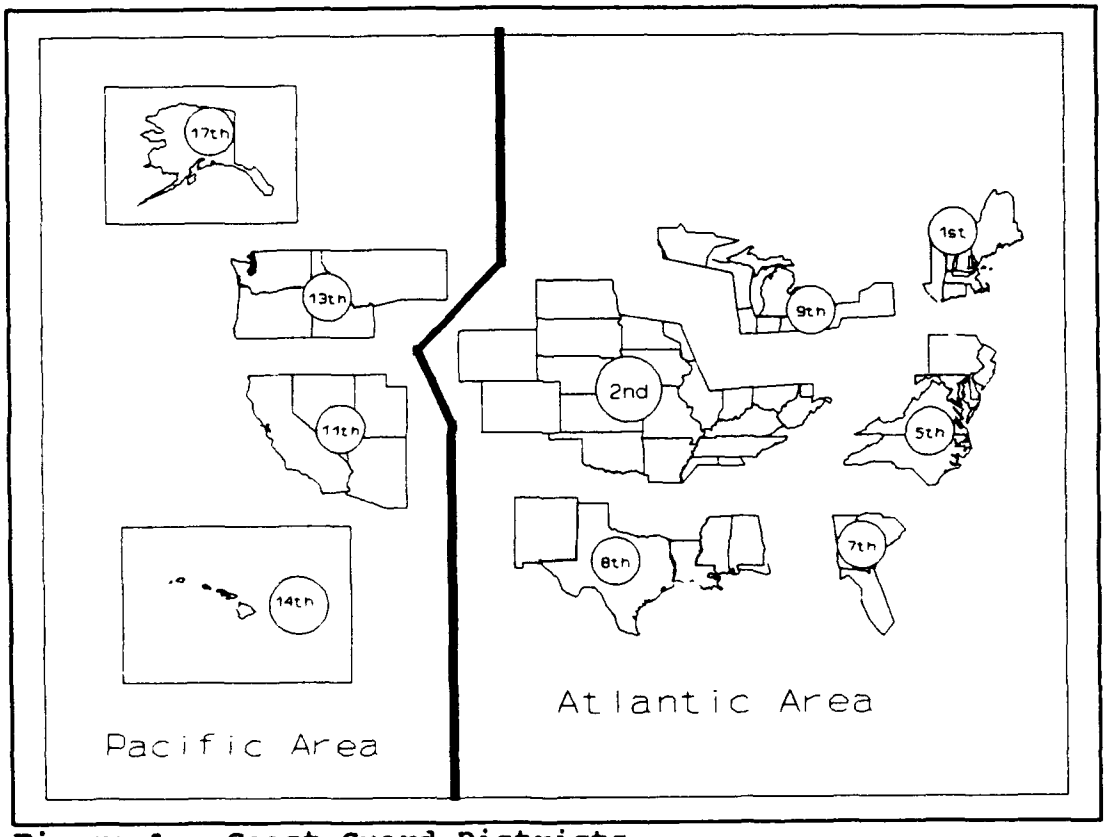

Figure 1: Coast Guard Districts

Coast Guard duties and responsibilities are historically organized into seven major operating programs --- Defense Readiness, Search and Rescue (SAR), Enforcement of Laws and

Treaties (ELT), Marine Environmental Protection (MEP), Aids to Navigation (ATON), Ice Operations (IOP), and Marine Safety. Each major operating program consists of one or more missions. Coast Guard missions are defined as "major subdivisions of Coast Guard statutory [operating programs] which are detailed in U.S. Law" [Ref. 2:p. 2]. To obtain "Coast Guard resource activity data for use in facility planning/management, program cost allocation and program management," the missions are further subdivided and expanded into approximately twenty employment categories [Ref. 3]. Employment categories are used to define the specific uses of a platform, including both training and operations. Table 1 provides an example of the relationship between major operating programs, missions, and employment categories as defined by the Coast Guard.

Although confusing and technically incorrect, it is common practice to refer to major operating programs as missions. Perhaps, this is due to the fact that the term mission is more readily identified with activities to be accomplished and, in essence, major operating programs are general groupings of these activities. This thesis will deal with requirements computed at the major operating program level. To avoid confusion, this thesis will differentiate programs and missions as defined in the previous paragraph or in Table 1. The abbreviations assigned in Table 1 will be used during this thesis and in the sample problem discussed in Chapter V.

TABLE 1.
MAJOR OPERATING PROGRAMS, MISSIONS AND EMPLOYMENT CATEGORIES

Major Operating Programs	Missions	Employment Categories
Marine Environmental Protection (MEP)	Marine Environmental Response	Marine Environmental Response
		Marine Inspection
Defense Readiness (MILOP)	Port Safety and Security	Port Safety and Security
	Defense Readiness	Military Ops
		Military Training
		Military Exercises
Enforcement of Laws and Treaties (ELT)	Fisheries Enforcement	Domestic Fisheries
		Foreign Fisheries
	Drug Interdiction	Drugs - Air Interdiction
		Drugs - Surface Interdiction
	Enforcement of Laws and Treaties	Immigration Laws
		ELT - Other
Aids to Navigation (ATON)	Short Range Aids to Navigation	Short Range Aids to Navigation
	Radio Navigation	Radio Navigation
Ice Operations (IOP)	Polar Operations	Polar Operations
	Domestic Ice	Domestic Ice
Marine Safety (MSAFE)	Commercial Vessel Safety	Marine Inspection Program
	Boating Safety	Boating Safety
Search and Rescue (SAR)	Search and Rescue	Search and Rescue

To accomplish all missions in the seven major operating programs, the Coast Guard owns and operates over 2,500 platforms consisting of approximately 300 cutters and over 200 fixed wing aircraft and helicopters. The remaining platforms are smaller vessels assigned to shore facilities (i.e., groups and stations) for performing near coastal and inland operations [Ref. 4:p. 150]. Most Coast Guard cutters and aircraft have a multi-mission capability to respond to "a wide variety of maritime activities and shift emphasis on short notice when need arises" [Ref. 5:p. 17]. This allows for greater flexibility in operations and procurement. However, this multi-mission capability of platforms also increases the complexity of assigning platforms to efficiently and effectively accomplish the missions in the seven operating programs on an annual basis. This problem of assigning platforms to missions, or directly to districts, is referred to as the force structure problem by the Coast Guard.

B. RELATED AND PRIOR WORK

In the past, the Coast Guard has attempted to quantify and solve the Force Structure problem, but lacked the necessary training or tools to analyze alternative solutions effectively [Ref. 6:p. 8]. The Coast Guard is currently without an acceptable analysis of its platform requirements (or force structure), though several projects are either being developed or proposed to assist in solving this formidable problem.

Most of the ongoing projects currently focus on specific missions and/or platforms. The following sections provide a brief description of relevant studies and ongoing projects.

1. Previous Coast Guard Studies

Prior to 1989 the Coast Guard annually developed two separate planning documents, cutter and aviation requirements, addressing the force structure problem. The purpose of these documents was to be:

> ...an analytical planning document which develops the Coast Guard's ... capital resources needed to achieve goals established in approved Operating Program Plans (OPP's)....[and] support formulation of the Coast Guard Capital Investment Projection, which assigns priorities over a five year time span [Ref. 7 & 8].

Though these two documents were independently prepared, they used similar methodologies. These methodologies are based on having individual program managers, one for each of the seven major operating programs, specify annual requirements in terms of employment hours for each district. To illustrate the concept of employment hours, consider a cutter which just completed one hour of an Enforcement of Laws and Treaties (ELT) mission. If the cutter monitors Search and Rescue (SAR) frequencies during part of the mission, the commanding officer may report that the cutter completed 1.5 employment hours, one hour for ELT and 0.5 hours for SAR, even though the actual mission time is only one hour.

There are two sources of data used in determining the number of required platforms for each district. One is from

the seven program managers who specify the number of employment hours to be performed by each platform type in order to accomplish missions in their programs. The other set of data is the hours required for platforms in each district to perform support programs such as public affairs, training and research and development. These hours are extracted from the data in the previous three years of the Abstract of Operations[2]. Summing the hours from the two sources gives the total employment hours required for each platform type in each district. In an attempt to compute the actual usage hours, the employment hours are multiplied by predetermined factors known as Simultaneous Mission Factors (or SMF) (see, e.g., Ref 7, page 2-4).

To determine the number of platforms required by each district, the Coast Guard simply divides the actual usage hours as computed above by the number of hours one platform can operate. If the number of required platforms of a particular type exceeds the number of platforms currently assigned to the district, then substitute platforms would be assigned to accomplish the hours instead.

In October 1989, the Cutter and Aircraft Rebaseline Requirements Study (or the Rebaseline Study) was completed and intended to replace the cutter and aviation requirement documents. The Rebaseline Study used a method similar to the

---

[2]A quarterly operations statistics report submitted by all Coast Guard platforms [Ref. 3]

ones in these documents. The main difference is that the total employment hours are based on estimates prepared by the members of the Rebaseline Study group, district and area commanders as well as the program managers.

Major drawbacks with the above studies are as follows. First, platforms selected or specified by program managers or the Rebaseline Study to perform a particular mission does not necessarily imply the efficient usage of the available platforms. This is due to the fact that some missions may be effectively performed by unselected platforms which may otherwise be under utilized. Second, specifying the number of mission employment hours required by each platform type in each district or area predetermines the number of platforms to be allocated to that district or area. Third, the concept of employment hours is subjective for it depends solely on the judgement of officers commanding the platforms [Ref. 3:p. 4-3]. Lastly, simultaneous mission factors which accurately convert employment hours to actual hours of usage are difficult to determine. These drawbacks may have undermined the acceptance of the Rebaseline Study.

2. Ongoing Projects

The Coast Guard has initiated two similar simulation projects to examine platform utilization in the performance of two major missions; SAR and ELT. Since June 1986, a KSS (knowledge-based decision support systems) project has been

under development to assist decision makers in the acquisition of cutters [Ref. 6]. Recently, a DSS (Decision Support System) was developed to examine the patrol boat (WPB)[3] Force Structure and Homeporting issues using simulation for the underlying models [Ref. 9]. The Coast Guard Research and Development (R&D) Center has requested Fiscal Years 92-93 funding for a "General Force-Mix" project "founded upon an optimization model" [Ref. 10]. Through these projects, the Coast Guard has gained a better understanding of the Force Structure problem and is beginning to use the decision tools made available through mathematical programming and management science techniques.

C. STUDY GOAL

The primary goal of this thesis is to develop a methodology to assist the Coast Guard in determining an effective platform allocation on an annual basis. This methodology involves modeling the allocation process as an optimization model and quantifying the mission requirements for each district.

It is hoped that the results described in this thesis illustrate the potential of mathematical programming models as decision aids and further encourage the Coast Guard to employ the methodology in the future.

---

[3]Patrol boats are Cutters 80 to 120 feet in length

D. THESIS ORGANIZATION

Chapter II describes a method of quantifying annual mission requirements and platform utilization. Chapter III outlines the formulation of an integer programming (optimization) model developed for the force structure problem. Chapter IV presents the implementation of this model using a commercially available Linear Programming solver and a research level front-end user interface. Chapter V provides a sample problem using Fiscal Year 1989 data with analysis of the model's results and the impact of modifications to the various parameters used to describe the platforms. Chapter VI summarizes the findings of this study and proposes topics for further study.

II. MISSION REQUIREMENTS AND PLATFORM UTILIZATION

As explained in Section B of Chapter I, previous Coast Guard projects or studies do not have a well defined methodology for quantifying mission requirements nor utilization of platforms. In this chapter, methods for quantifying these two factors in planning Coast Guard force structure are proposed. The main objective is to provide a reproducible and objective method for quantifying or estimating annual requirements for the seven major operating programs and specifying the usage of various platforms. Here, reproducibility means that, given the same set of data and using the proposed methods, different decision makers should be able to produce the same annual requirements and platform utilization.

In addition, to gain acceptance in the Coast Guard community, new definitions or data requirements are kept to a minimum. Many parameters are provided to allow incorporation of judgmental factors which are best left to decision makers. In fact, officials at the Coast Guard headquarters in Washington, D.C. are kept abreast and express no objection to the proposed methodology.

A. QUANTIFYING ANNUAL MISSION REQUIREMENTS

In the same major operating program, missions are generally different depending on the distance which must be traveled by the platform types, the length of the mission, the weather conditions, etc. Thus, it would be insufficient to simply specify total annual hours of platform usage for each operating program. When considering the fact that one platform may be capable to perform several types of missions, this insufficiency is more evident. For example, a platform designed primarily for ELT may not be able to perform a SAR mission at the same level of effectiveness and efficiency as a platform which is primarily designed for SAR. So, only specifying annual mission hours and allowing for multiple mission capability implies perfect substitution between platforms primarily designed for different types of missions.

To allow for the difference in missions in the same program and more realistic degree of substitution, missions are furthered classified into three different classes. The choice and definition of the three classes are based on ten years of personal experience in the Coast Guard. However, they also serve to illustrate the basic concept. Decision makers with more experience and knowledge are more suitable to make the final determination. The implementation in Chapters IV and V also allow users to define their own classifications.

Table 2 and Table 3 display the classification of mission classes for surface and air platforms respectively. Among the three classes, Class I is the easiest and Class III is the hardest. Thus, missions in Class III should be mainly assigned to platforms specifically designed for the more difficult operating environment. On the other hand, missions in Class I may be performed by most if not all platforms in the Coast Guard inventory. The numerical attributes (Distance, Endurance, Speed, and Weather) are based on personal experience and they should be fully agreed upon by decision makers before full implementation.

TABLE 2.
MISSION CLASSES FOR SURFACE PLATFORMS

	Class I	Class II	Class III
Distance off shore	less than 100nm	100 - 200 nm	200+ nm
Endurance	1 week	1 - 2 weeks	2+ weeks
Speed	at least 10kts	10kts - 20kts	20+ kts
Weather	Good	Fair	Poor

TABLE 3.
MISSION CLASSES FOR AIR PLATFORMS

	Class I	Class II	Class III
Distance off shore	less than 150nm	150 - 300nm	300 - 400nm
Endurance	3 hrs	4 hrs	5 hrs
Weather	Good	Fair	Poor

With this classification, the mission hours required by each major program are further subdivided into three classes, e.g., Class I-SAR, Class II-SAR and Class III-SAR. The next section then describes how to allocate time of each platform to each major operating program and class.

B. PLATFORM UTILIZATION

Although platforms may be multi-mission capable, they are generally designed to serve a specific program in a particular operating environment. Thus, care must be taken to prevent the over use of platforms in programs and operating environments not intended for them. Coast Guard documents such as the Sponsor's Requirements Document (SRD) and Commandant Instruction 3501.26 (series) serve as a foundation in establishing limits for the utilization of platforms to perform various missions in the three different classes.

The Sponsor's Requirements Document details capabilities and limitations of platforms proposed for construction. Existing and new platforms must all have a SRD prepared prior to construction. It is based on this SRD that one can specify the maximum percent utilization of a given platform type in the three different classes of missions. Table 4 provides examples for maximum percent utilization for several platform types.

TABLE 4.
CLASS UTILIZATION LIMITS

PLATFORM	CLASS I	CLASS II	CLASS III
82' WPB	100%	50%	10%
180' WLB	100%	100%	50%
378' WHEC	100%	100%	100%

The first row of Table 4 gives utilization limits for 82' WPB which is a patrol boat. From the limits, it can be ascertained that this patrol boat is primarily designed for mission of Class I type for it is allowed to use all (100%) of its available operating hours. However, it can only use 50% and 10% of its hours to perform missions in Class II and III respectively. Note that these percentages represent the maximum utilization in the three different classes; so they do not have to sum to 100%.

Commandant Instruction 3501.26 (series) provides data concerning the Required Operational Capabilities (ROC) of various platforms. The format of this data is a table which lists the primary and secondary major operating programs suitable for each platform type. However, reviewing historical Abstract of Operations data reveals that there were platforms which performed missions not in their primary or secondary programs. This prompted an extension of the ROC categories for assigning maximum percent utilization of platforms to various programs. The basic idea is to allow a

high degree of utilization in the primary and secondary programs of each platform type. However, limited utilization would be allowed for missions not included in the primary or secondary programs. Table 5 gives examples of maximum percentage assignment for several platforms.

TABLE 5.
PROGRAM UTILIZATION LIMITS

Platform	SAR	ELT	ATON	IOP	MEP
82' WPB	80%	100%	10%	0%	0%
180' WLB	40%	50%	100%	50%	10%
378' WHEC	80%	100%	0%	0%	20%

It can be gathered from the first row that ELT and SAR are the primary and secondary missions for an 82' patrol boat. The remaining 10% limit for ATON represent the utilization found in the historical Abstract of Operations data. Other programs with zero percent utilization limits are simply not suitable for an 82' WPB.

Multiplying the class and program utilization tables together would provide the maximum percent utilization of a platform type by program and class. Although not explicitly stated, it should be clear that, prior to full implementation of the method described in this thesis, the percentages should be approved by all concerned parties in the Coast Guard.

Percentages used in this thesis are based on personal experience and offered only as an illustration.

Chapter III describes the formulation of this problem as a mixed integer linear programming model.

III. MATHEMATICAL MODEL

This chapter describes a mathematical programming model which allocates platforms to districts and local area commands to satisfy the annual mission requirements as quantified in Chapter II. The model attempts to use currently available platforms to satisfy the requirements. When necessary, the relocation of platforms from one district to another is permitted. However, the operational organization precludes the relocation of area platforms to districts. In addition, the model is also permitted to acquire new platforms if the available platforms are insufficient. It should be pointed out that it is not intended for the Coast Guard to purchase new platforms when the model's answer so states. Instead, allowing for new platforms is simply a tool to prevent model infeasibility and points out deficiency in the current platform availability. The decision to acquire new platforms requires models which allow for, e.g., multiple periods and service life extension programs. Models of this type are generally large and require data not yet available. Hence, consideration of such models is recommended for future investigation.

A. LINEAR MIXED INTEGER PROGRAM MODEL FORMULATION

Indices:

$p = 1,\ldots,P$		platforms
$d,\delta = 1,\ldots,D$		districts
$m = 1,\ldots,M$		missions
$l = 1,\ldots,3$		class

Index Sets:

Ω	district surface platforms	$\Omega \subseteq \{1,\ldots,P\}$
Φ	district air platforms	$\Phi \subseteq \{1,\ldots,P\}$
Γ	area surface platforms	$\Gamma \subseteq \{1,\ldots,P\}$

where $\{1,2,\ldots,P\} = \Omega \cup \Phi \cup \Gamma$ and Ω, Φ and Γ are mutually disjoint.

Given and Derived Data:

$u^p_{m,l}$	upper utilization limits for hours expended on mission m at class l by platform p (see Chapter II, Section B)
hr^p	maximum number of hours platform p can perform per year
c^p	annual operating cost for platform p
$sh_{d,m,l}$	annual surface hours required in district d for mission m at class l (see Chapter II, Section A)
$ah_{d,m,l}$	annual air hours required in district d for mission m at class l (see Chapter II, Section A)
sp_m	maximum percentage of air hours required which can be performed by surface platforms
ap_m	maximum percentage of surface hours required which can be performed by air platforms
E^p_d	current number of platform p located in district d

$F^{p}_{m,l}$ factor for converting air hours to surface hours for mission m class l and by air platform p where $p \in \Phi$

$G^{p}_{m,l}$ factor for converting surface hours to air hours for mission m class l and by surface platform p where $p \in \Omega \cup \Gamma$

a^{p} cost of acquiring a new platform type p

r^{p} penalty for relocating platform p

Nonnegative Variables:

$ZA^{p}_{d,m,l}$ number of surface hours performed by platform p in district d for mission m at class l, where $p \in \Phi$ (see Figure 2)

$ZS^{p}_{d,m,l}$ number of air hours performed by platform p in district d for mission m at class l, where $p \in \Omega \cup \Gamma$ (see Figure 2)

$Y^{p}_{d,m,l}$ number of hours platform p in district d performs mission m at class l (see Figure 2)

$\Delta^{p}_{d,\delta}$ number of district platforms p moved from district d to district δ where $p \in \Omega \cup \Phi$ (see Figure 3)

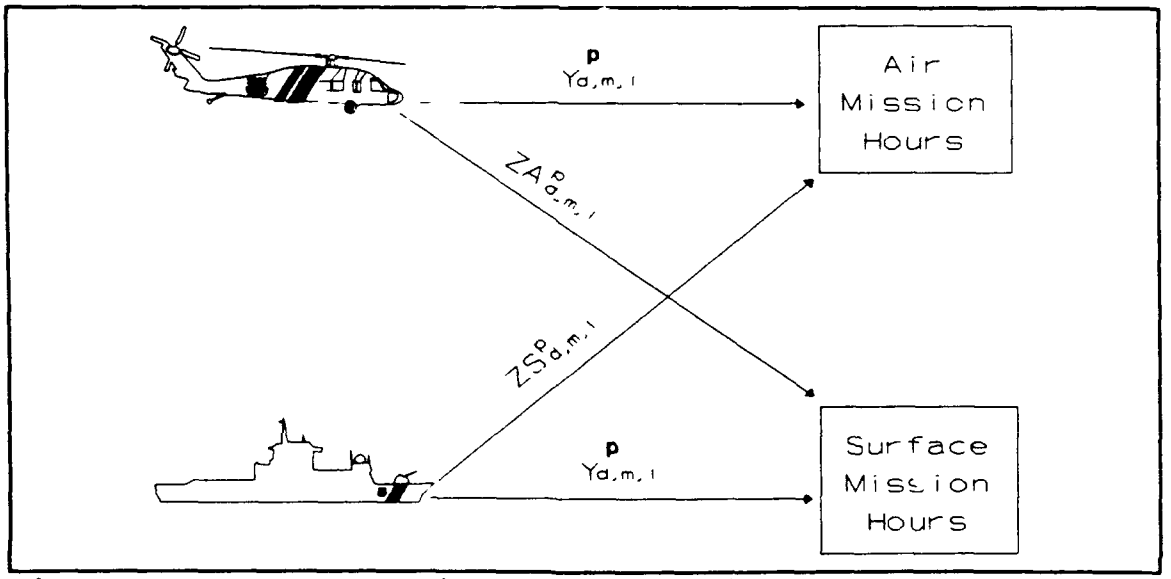

Figure 2. Representation of Hour Variables

Nonnegative Integer Variables:

OXA^p number of currently available (old) area surface platforms assigned where $p \in \Gamma$

OXD^p_d number of currently available (old) district platforms p assigned to district d where $p \in \Omega \cup \Phi$

NXA^p number of new area surface platforms assigned though not currently in inventory where $p \in \Gamma$

NXD^p_d number of new district platforms p assigned to district d but not currently in inventory, where $p \in \Omega \cup \Phi$

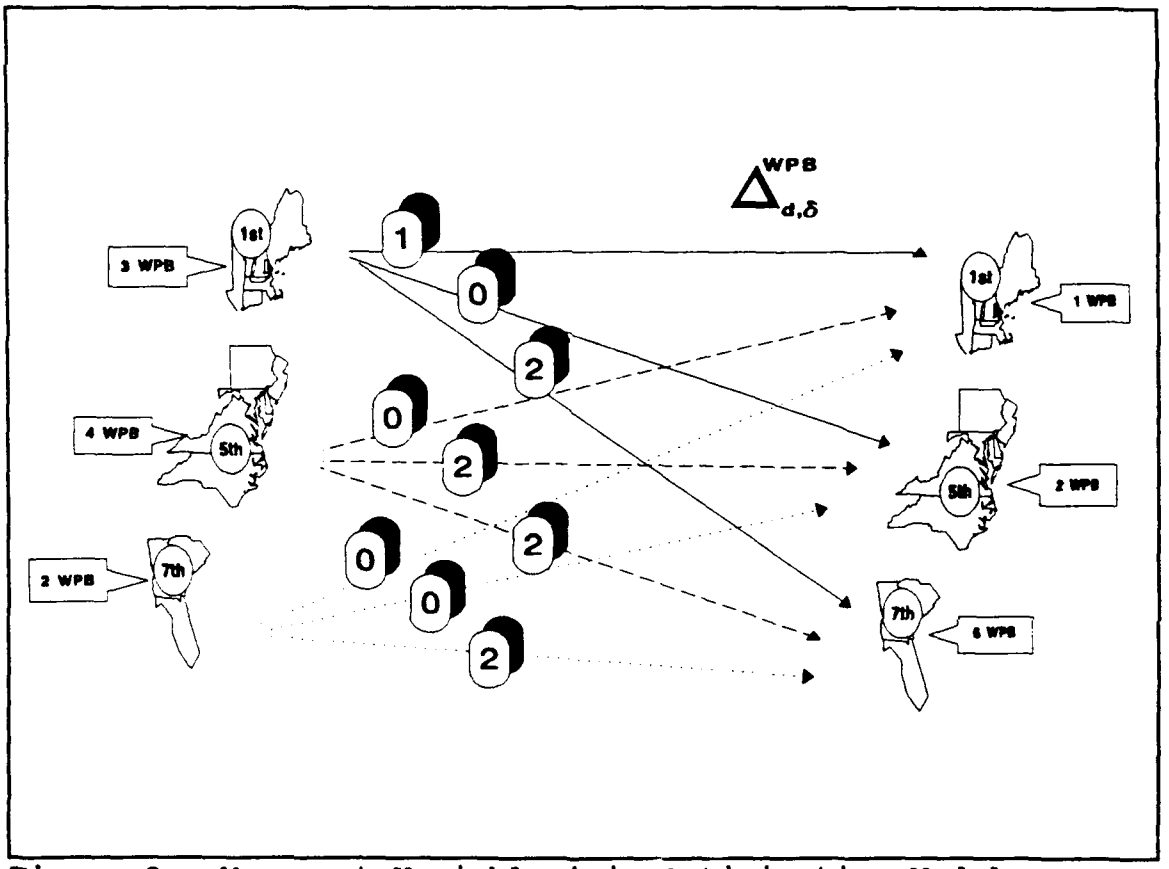

Figure 3. Movement Variable Δ in Optimization Model

Formulation:

$$MINIMIZE \sum_{d=1}^{D} \sum_{p \in \Omega \cup \Phi} c^p \left(OXD_d^p + NXD_d^p \right) + \sum_{p \in \Gamma} c^p \left(OXA^p + NXA^p \right)$$
$$+ \sum_{d=1}^{D} \sum_{p \in \Omega \cup \Phi} a^p \, NXD_d^p + \sum_{p \in \Gamma} a^p \, NXA^p$$
$$+ \sum_{d=1}^{D} \sum_{\delta=1}^{D} \sum_{p \in \Phi \cup \Omega} r^p \, \Delta_{d,\delta}^p$$

Subject to:

$$\sum_{p \in \Omega \cup \Gamma} Y_{d,m,l}^p + \sum_{p \in \Phi} F_{m,l}^p \, ZA_{d,m,l}^p \geq sh_{d,m,l} \quad \forall \; d,m,l \tag{1}$$

$$\sum_{p \in \Phi} Y_{d,m,l}^p + \sum_{p \in \Omega \cup \Gamma} G_{m,l}^p \, ZS_{d,m,l}^p \geq ah_{d,m,l} \quad \forall \; d,m,l \tag{2}$$

$$\sum_{l=1}^{3} \sum_{p \in \Phi} ZA_{d,m,l}^p \leq ap_m \sum_{l=1}^{3} sh_{d,m,l} \quad \forall \; d,m \tag{3}$$

$$\sum_{l=1}^{3} \sum_{p \in \Omega \cup \Gamma} ZS_{d,m,l}^p \leq sp_m \sum_{l=1}^{3} ah_{d,m,l} \quad \forall \; d,m \tag{4}$$

$$\sum_{m=1}^{M} \sum_{l=1}^{3} Y_{d,m,l}^p + \sum_{m=1}^{M} \sum_{l=1}^{3} ZS_{d,m,l}^p \leq hr^p \left(OXD_d^p + NXD_d^p \right) \quad \forall \; d, p \in \Omega \tag{5}$$

$$\sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{l=1}^{3} Y_{d,m,l}^{p} + \sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{l=1}^{3} ZS_{d,m,l}^{p} \le hr^{p}\left(OXA_{d}^{p} + NXA_{d}^{p}\right) \quad \forall\, p \in \Gamma \qquad (6)$$

$$\sum_{m=1}^{M}\sum_{l=1}^{3} Y_{d,m,l}^{p} + \sum_{m=1}^{M}\sum_{l=1}^{3} ZA_{d,m,l}^{p} \le hr^{p}\left(OXD_{d}^{p} + NXD_{d}^{p}\right) \quad \forall\, d,p \in \Phi \qquad (7)$$

$$Y_{d,m,l}^{p} + ZS_{d,m,l}^{p} \le hr^{p}\left(OXD_{d}^{p} + NXD_{d}^{p}\right)u_{m,l}^{p} \quad \forall\, d,m,l,p \in \Omega \qquad (8)$$

$$Y_{d,m,l}^{p} + ZA_{d,m,l}^{p} \le hr^{p}\left(OXD_{d}^{p} + NXD_{d}^{p}\right)u_{m,l}^{p} \quad \forall\, d,m,l,p \in \Phi \qquad (9)$$

$$\sum_{d=1}^{D} Y_{d,m,l}^{p} + \sum_{d=1}^{D} ZS_{d,m,l}^{p} \le hr^{p}\left(OXD_{d}^{p} + NXD_{d}^{p}\right)u_{m,l}^{p} \quad \forall\, m,l,p \in \Gamma \qquad (10)$$

$$\sum_{\delta=1}^{D} \Delta_{d,\delta}^{p} \le E_{d}^{p} \qquad \forall\ d,\ p \in \Omega \cup \Phi \qquad (11)$$

$$\sum_{d=1}^{D} \Delta_{d,\delta}^{p} = OXD_{\delta}^{p} \qquad \forall\ \delta,\ p \in \Omega \cup \Phi \qquad (12)$$

$$OXA^{p} \le \sum_{d=1}^{D} E_{d}^{p} \qquad \forall\ p \in \Gamma \qquad (13)$$

The objective function consists of three terms. The first term computes the annual operating cost of the platforms utilized by the model. The second and third terms calculate the acquisition and relocation cost, respectively.

Constraints (1) and (2) simply ensure that the number of hours performed by the various platforms meet the annual surface and air mission requirements, respectively. It should be noted that both constraints allow surface platforms to perform missions normally assigned to air platforms and vice versa. However, this cross utilization of platforms is not allowed to exceed certain percentages specified by the user. Constraint (3) then limits the number surface mission hours to be performed by air platforms to be no more than a percentage (ap_m) of the required surface mission hours. Similarly, constraint (4) places a limit on surface platforms.

Constraints (5) through (7) ensure that there are enough platforms in each district and/or area to fulfill the hours allocated to perform various mission requirements as specified by constraints (1) through (4). Constraints (8) through (10) guarantee that the upper utilization limits, as defined by Tables 4 and 5 in Chapter II, are not violated. These limits restrict the maximum number of hours that district and area platforms can perform missions in the three classes.

Constraints (11) and (12) together account for the relocation of existing platforms. Constraint (11) ensures that the number of platforms relocated from district d to other districts does not exceed the number of platforms originally stationed at d. Constraint (12) then guarantees that the number of platforms at district δ, relocated from other districts or otherwise, satisfies the allocation, OXD_{δ}^{P},

specified in constraints (5) - (8). Constraint (13) is included to ensure that the number of actual area platforms assigned does not exceed the current available number of area platforms. All platforms which exceed the current force structure must be considered new and be charged with the acquisition cost.

The above model is basic to the force structure problem. A variety of embellishments are possible when additional data becomes available. As an example, constraints (14) and (15) below can be appended to the basic model to guarantee that the budgets for acquisition of new platforms and relocation of existing ones are not exceeded.

$$\sum_{d=1}^{D} \sum_{p \in \Omega \cup \Phi} a^p \, NXD_d^p + \sum_{p \in \Gamma} a^p \, NXA^p \leq alim \qquad (14)$$

$$\sum_{d=1}^{D} \sum_{\delta=1}^{D} \sum_{p \in \Omega \cup \Phi} r^p \, \Delta_{d,\delta}^p \leq mlim \qquad (15)$$

In this case, the variables alim and mlim represent the two budgets.

Chapter IV describes the implementation of this model and the development of a research level front-end user interface to facilitate in solving the force structure problem using the model developed above.

IV. IMPLEMENTATION AND USER INTERFACE

The primary considerations for the selection of a LP solver and the programming language for the interface were portability and the current availability of software within the Coast Guard. The mathematical model described in Chapter III was implemented using the General Algebraic Modeling System (GAMS) [Ref. 13] available on the Amdahl 5990-500 computer at the Naval Postgraduate School. To facilitate the input and modification of data for the model, a research level front-end user interface was written in FORTRAN. This interface allows for ease of data input and shields users away from the mathematical model implemented in GAMS. As a byproduct, the interface also helps in maintaining the data since it has many of the basic functions of a database system. The description of this interface is fully described below.

A. USER INTERFACE

To allow ease of use, an interface was developed to make the GAMS model transparent to the user. It is assumed that the user of this interface would be knowledgeable about the force structure problem but would not need to know the intricacies of the GAMS formulation or optimization techniques. The data required for this model is primarily

constant and should be store in data files to avoid needless repetitive data entry by the user. Storing and retrieving the model's data via files reduces the chance of data entry errors thus increasing the efficiency of the system and ultimately enhancing user acceptance. However, all data should be readily available to the user for modification as needed. Figure 4 provides a overview of the relationship between the data, the interface and GAMS.

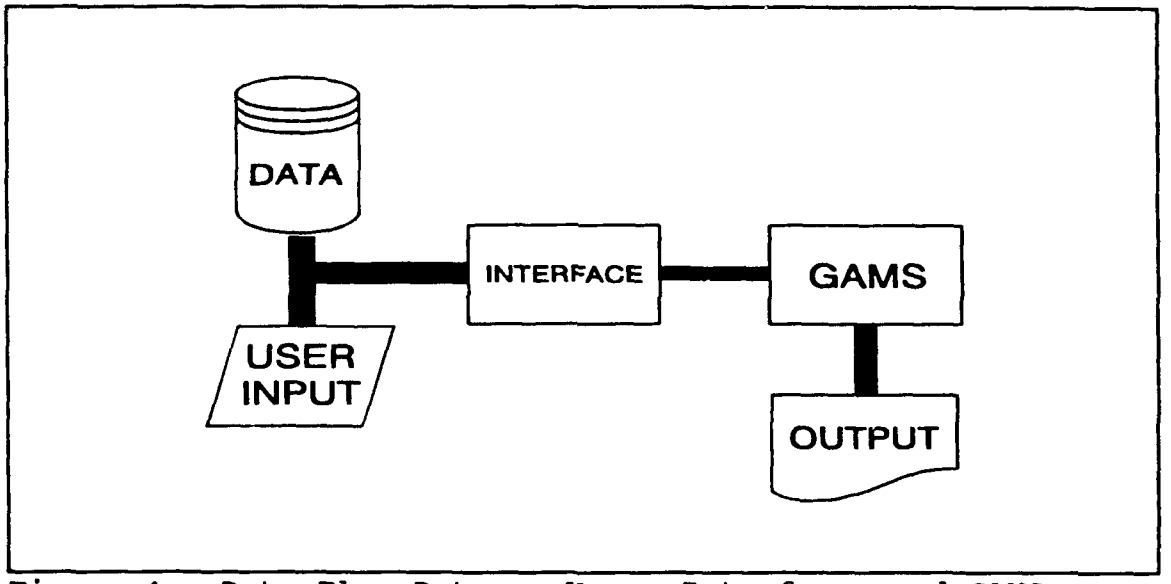

Figure 4: Data Flow Between User, Interface, and GAMS

The interface is menu driven with a main menu which provides the user with access to all the functions shown in Figure 5. Most of these functional subroutines provide the user with different menus to further define the desire of the user.

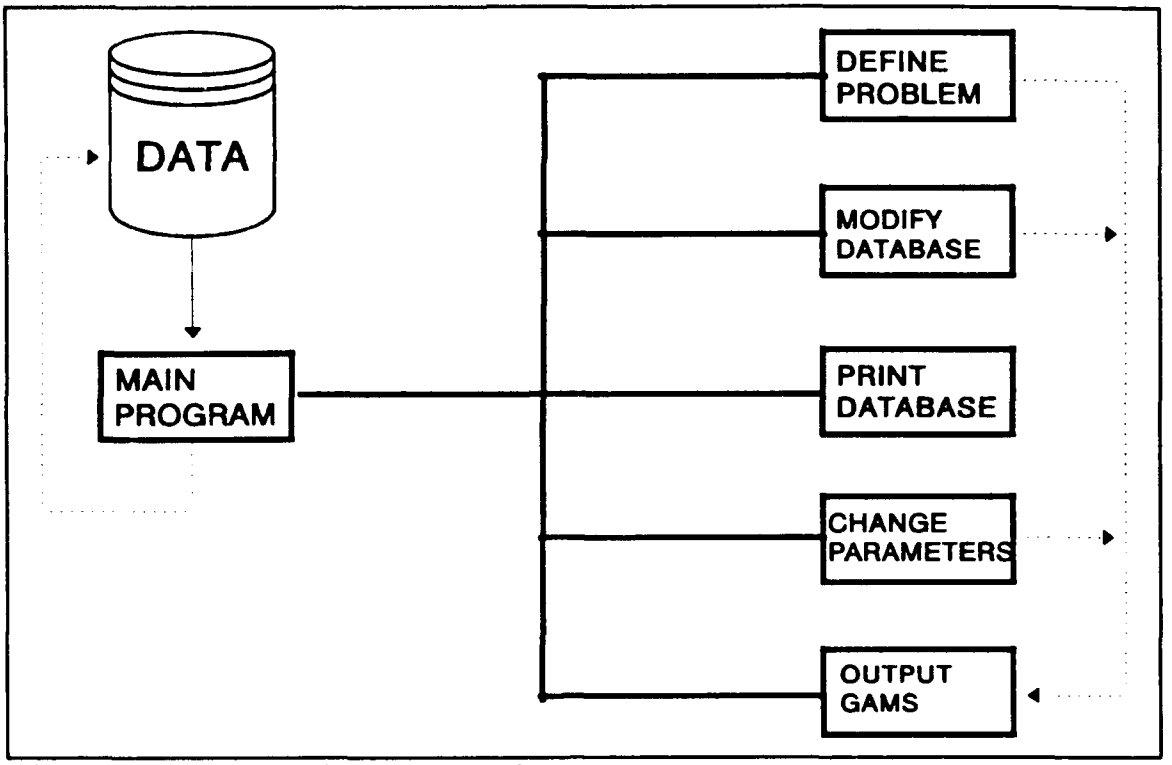

Figure 5: Functional Routines of User Interface

The user interface is a simple database program which allows modification of the data base, additions and deletions and the like. Additionally, the user must be able to view the database and produce output in either an easily understood format or in the format required by GAMS to solve the model. Though these two formats are not considered mutually exclusive, the ability to provide concise summary tables of the data for the user is preferable over printing out the sets, parameters, and tables used by the GAMS formulation. Figures 6 and 7 provide samples of the formatted output available from the interface's PRINT DATABASE option.

PLATFORM NAME: WPB110
PLATFORM TYPE: DISTRICT SURFACE

YEARLY OPERATING DATA:
COST($M): 0.590 HOURS: 2000

MISSION PROFILE DATA

MISSION	DESCRIPTION	PERCENTAGE
SAR	SECONDARY MISSION	50%
ELT	PRIMARY MISSION	100%
ATN	NOT ASSIGNED	0%
IOP	NOT ASSIGNED	0%
MEP	TERTIARY MISSION	10%
MSAFE	TERTIARY MISSION	10%
MILOP	SECONDARY MISSION	50%

CLASS PROFILE DATA

CLASS	DESCRIPTION	PERCENTAGE
1	FULLY CAPABLE	100%
2	PARTIALLY CAPABLE	50%
3	MARGINALLY CAPABLE	10%

CURRENT LOCATION OF ALL WPB110 PLATFORMS:

FIRST 2 FIFTH 2 SEVENTH 16
EIGHTH 2 NINTH 0

Figure 6: Sample Platform Database Output

DISTRICT: FIRST

	SURFACE HOURS REQUIRED CLASS			AIR HOURS REQUIRED CLASS			
MISSION	1	2	3	1	2	3	TOTAL
SAR	5000	1000	65	668	800	400	7933
ELT	8000	6000	1416	800	900	2336	19452
ATN	6000	3000	125	200	44	0	9369
IOP	800	434	200	0	0	4	1438
MEP	500	48	0	328	200	0	1076
MSAFE	500	50	0	15	0	0	565
MILOP	1500	1000	250	800	200	133	3883
TOTAL	22300	11532	2056	2811	2144	2873	43716

CURRENT PLATFORMS LOCATED WITHIN FIRST DISTRICT BOUNDARIES:

NAME	QTY	NAME	QTY	NAME	QTY	NAME	QTY
HU25	5	HH65	5	HH60	4	C130	0
WPB82	8	WPB110	2	WTGB	3	WLB180	3
WLM157	2	WLM133	3	WHEC	1	WMEC270	2
WMEC210	1						

Figure 7: Sample District Database Output

The output is designed to provide the necessary information to the various decision makers involved with the Coast Guard force structure problem. The operational commanders would be most interested in the summary data concerning their district, the program managers would likewise be concerned with the data pertaining to the individual missions. The data displayed in Figures 6 and 7 must be formatted slightly differently to meet the structure of sets, tables and parameters required by GAMS. Figure 8 provides an example of the interface's ability to convert the database

into the data structures needed by GAMS for proper model definition and formulation.

```
SETS
    P   ALL PLATFORMS
      / WPB82   ,WPB110  ,WTGB    /

    IDS(P)  DISTRICT SURFACE PLATFORMS
      /   WPB82   ,  WPB110,  WTGB  /

    D  DISTRICT
      / FIRST   ,FIFTH   ,SEVENTH  ,EIGHTH  ,NINTH /

    M  MISSION
      / SAR  ,ELT  ,ATN  ,IOP  ,MEP  ,MSAFE  ,MILOP/

TABLE
  ADS(IDS,M)  MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IDS)
CAN
              PERFORM MISSION (M).
                  SAR   ELT   ATN   IOP   MEP MSAFE MILOP
        WPB82     1.00  1.00  0.00  0.00  0.10  0.10  0.00
        WPB110    0.50  1.00  0.00  0.00  0.10  0.10  0.50
        WTGB      1.00  0.50  0.00  1.00  0.10  0.50  0.10;

TABLE
  CURRENT(P,D)  CURRENT FORCE STRUCTURE FOR THE COAST GUARD
          FIRST      FIFTH      SEVENTH      EIGHTH      NINTH
  WPB82     8                      9           10          0
  WPB110    3          2          17           2          0
  WTGB      3          1          0                      5;

PARAMETERS
   RHDS(IDS)  MAXIMUM NUMBER OF RESOURCE HOURS PLATFORM (IDS)
              CAN PERFORM PER YEAR.
                 /   WPB82         1650
                     WPB110        2000
                     WTGB          2300  /

   COSTDS(IDS)  OPERATING COSTS FOR PLATFORM (IDS) PER YEAR IN
                MILLIONS OF DOLLARS.
                  /   WPB82        0.370
                      WPB110       0.590
                      WTGB         0.650 /;
```

Figure 8: Interface Output Following GAMS Data Structure

Though the GAMS model representation is purported to be easily read by people and by computers [Ref. 13], the alternate output available through the interface provides the necessary information in a more compact and understandable

format for the user concerned more with the problem than with the formulation of the model.

To prevent users from accidentally providing data for which no feasible solution exists, the interface performs a simple preprocessing of the entered data. This preprocessing checks that there is at least one platform type capable of performing each mission-class pair for which there is a positive mission hour requirement. Other forms of infeasibility would be detected by GAMS.

B. GENERAL ALGEBRAIC MODELING SYSTEM

The Coast Guard has recently acquired GAMS software for ongoing research into the force structure problem. The decision to implement this model using GAMS was motivated by the designed portability of the software [Ref. 13:p. 4] and the Coast Guard's recent acquisition of this particular system. Another principle of GAMS design is that "the optimization problem should be expressible independently of the data it uses" [Ref. 13:p. 4]. Thus GAMS allows for the separation of the model's data from the actual formulation of the model. The data and formulation can be physically separated with the use of the INCLUDE statement. This feature allows the data for the model to be stored in a different file for later use with a GAMS formulation stored in another file. This also allows for numerous data files to be developed and

saved for use with the model while retaining only one copy of the model formulation and associated output statements.

The FORTRAN-based solver called XMP/ZOOM [Ref. 13 & 14] is the only mixed integer programming solver currently available with the Coast Guard version of GAMS. By necessity, XMP/ZOOM was used as the integer programming solver for this mathematical model. During the development and testing of the mathematical model several problems became apparent with the use of XMP/ZOOM which is partly due to the fact that ZOOM "is intended for medium-sized problems with no special structure and up to about 200 zero/one variables" [Ref. 13:p. 225].

Chapter V will discuss the computational difficulties experienced with XMP/ZOOM and examine the feasibility of the optimization model by using Fiscal Year 1989 data mission hour requirements and platform availability.

V. SAMPLE PROBLEM

After verifying the correctness of the model with a small data set, a realistic data set was considered in order to examine the operating characteristics of the model and its interface for the anticipated utilization by the Coast Guard. The sections below contain a description of how the data set was obtained, a discussion concerning the performance of the model and its solver, and sample analyses of the model outputs.

A. PROBLEM DATA

As stated in Chapter I, the Coast Guard organization is divided into two area commands, Atlantic and Pacific. Since they operate independently in that they do not share platforms, only the force structure problem for the Atlantic area command is considered. However, the Pacific area command can be modeled in a similar manner. The Atlantic area command contains six districts: First, Second, Fifth, Seventh, Eighth, and Ninth. The Second district consists of states in the midwest and is not adjacent to any coastal waters. This district employs only one type of platform, river buoy tenders to perform virtually one type of mission, ATON on the riverways. So, the problem of determining the number of

platforms for the Second district is simple and can be considered independently from the other districts in the Atlantic area command. Thus, the model contains one area command and five districts.

Since this thesis only considers the force structure at the district and area levels, smaller platforms (vessels less than 65 feet in length) are excluded from the data set. Also excluded are icebreakers, construction tenders and the training vessel Barque Eagle. These platforms are used primarily for single missions and can be considered independently. Thus, the platform types included in this study are cutters and aircraft noted for their multi-mission design. The operating cost for these platforms were estimated from three sources: FY89 Standard Personnel Cost tables prepared by the Coast Guard Budget Division, cost tables provided by the Aviation Branch of the Coast Guard Operations Division, and discussions with Coast Guard officials. Additionally, to define utilization limits as described in Chapter II, the capabilities, design specifications, and manning levels for cutters and aircraft were extracted from Commandant Instruction 3501.26, The Ships and Aircraft of the U.S. Fleet, and personal experience.

The required mission hours were extracted from the FY1989 Abstract of Operations [Ref. 12 & 15]. Using information inferred from this document and personal experience, these mission hours were separated into the three mission classes as

defined in Chapter II. Mission hours in this data set include those hours utilized for missions benefiting the seven major operating programs; these hours do not include those used for training. To account for the required training hours, the available operating hours of each platform type are reduced by the number of hours needed for training (see Ref. 11). Training is required for every platform and consumes a fairly constant amount of time. Therefore, subtracting the training hours from the available hours of each platform is not only a simple solution, but also prevents the model from assigning all training hours to, e.g., a single platform.

Finally, acquisition costs and movement costs for the various platform types were set to constants, since these costs are not currently available. It should be emphasized that the data used in this sample problem is for illustrative purposes only. For example, required mission hours extracted from the Abstract of Operations would not be appropriate for actual planning, since these reports only document the mission hours performed by various platforms. In general, the total hours performed are restricted by the availability of platforms. When platforms are unavailable, some required missions are left unfulfilled and therefore not recorded in the Abstract of Operations. Thus the hours reported in this document underestimate the actual mission hours needed.

B. SOLVING THE MODEL

During the development and testing of this model it was found that XMP/ZOOM consumed vast amounts of computer resource time while attempting to optimally solve the model. Given the uncertainty involved in determining future mission needs and the inaccuracy in forecasting operating costs for newly constructed platforms [Ref. 16], a near optimal solution, e.g., within twenty percent, would be acceptable. However, a ten percent or less tolerance is desirable.

To further reduce the computational time to an acceptable level, the following cutting plane is added to the model.

$$\sum_{d=1} \sum_{p \in \Omega \cup \Phi} \left(OXD_d^p + NXD_d^p \right) + \sum_{p \in \Gamma} \left(OXA^p + NXA^p \right) \geq T \qquad (16)$$

where T is the *floor* of the total number of platforms obtained by solving the model without the integrality restriction (or the relaxed model). Theoretically, this cutting plane is not valid. However, based on the results in Table 6, the model with this cut closely approximated the original model.

TABLE 6
SOLUTIONS TO RELAXED MODELS WITH AND WITHOUT THE CUT

Model Version	Model w/o the Cut	Model with the Cut	Relative Difference
I	165.5081	165.9127	0.24%
II	158.6821	158.7665	0.05%
III	163.4569	163.5397	0.05%
IV	169.1057	169.1797	0.04%

Table 6 reports the objective function values for four versions of the relaxed model with and without the cut. Since the two sets of values do not differ significantly it is assumed that the feasible region of the integer model with and without the cut are equivalent or nearly so.

Without the cut, version II of the model consumed over five hours of CPU time and still failed to reach an integer solution. With the cut, the model used 30 minutes of CPU time to produce an integer solution within 15 percent of optimality. On the average, models with the cut took approximately 23 minutes to obtain an acceptable (within 15 percent) integer solution. Thus, to perform sample analyses, the cut was added to each version of the model discussed below.

C. OUTPUTS AND ANALYSIS

Figure 9 displays the current locations of all district and area platforms used for all runs mentioned in this section. These locations were obtained, as before, from FY1989 Abstract of Operations. To insure that data in Figure 9 and those gathered in Section B yield a meaningful problem, the model in Chapter III was solved with the cross utilization factors of platforms set to zero. This disallows the surface platforms to perform air missions and vice versa. The resulting model contains 2437 continuous variables, 106 discrete variables, and 1182 equations and took 1,758 CPU

seconds to obtain a solution within seven percent of optimality. The result from the ZOOM solver showed that the data set is meaningful for it generates a feasible problem.

PLATFORM	DISTRICT						
	1ST	5TH	7TH	8TH	9TH	AREA	TOTAL
AIRCRAFT							
HU25	5	0	10	3	0	-	18
HH65	5	3	16	11	4	-	39
HH3	4	3	9	0	2	-	18
C130	0	5	8	0	0	-	13
SURFACE							
WPB82	8	8	8	10	0	-	34
WPB110	2	2	16	2	0	-	22
WTGB	3	1	0	0	5	-	9
WLB180	3	3	2	2	5	-	15
WLM157	2	3	0	0	0	-	5
WLM133	3	0	1	2	0	-	6
WHEC	-	-	-	-	-	1	1
WMEC270	-	-	-	-	-	10	10
WMEC210	-	-	-	-	-	10	10

Figure 9: Fiscal Year 1989 Coast Guard Force Structure

Figures 10 - 12 are part of the outputs from the model. A full listing of model outputs are provided in Appendix C. Figure 10 shows how the model reallocates the platforms. Recall that area platforms can not be relocated to districts and vice versa. So, area platforms are not relocated and district platforms must be relocated among themselves. It is interesting to note that the model does not utilize all available platforms. Figure 11 displays the model allocation of the operating hours of each platform type to each major operating program. To provide a more detail allocation of these operating hours, the model also provides the allocation

by districts. For example, Figure 12 illustrates the allocation for the seventh district.

PLATFORM	DISTRICT						
	1ST	5TH	7TH	8TH	9TH	AREA	TOTAL
AIRCRAFT							
HU25	1	1	2	0	0	-	4
HH65	7	6	16	8	2	-	39
HH3	4	0	12	0	0	-	16
C130	0	1	2	0	1	-	4
SURFACE							
WPB82	8	6	15	5	0	-	33
WPB110	2	2	16	2	0	-	22
WTGB	3	1	1	0	4	-	9
WLB180	0	0	1	2	2	-	5
WLM157	2	3	0	0	0	-	5
WLM133	1	1	2	2	0	-	6
WHEC	-	-	-	-	-	0	0
WMEC270	-	-	-	-	-	5	5
WMEC210	-	-	-	-	-	10	10

Figure 10: Model's Proposed 1989 Coast Guard Force Structure

PLATFORM	MISSION						
	SAR	ELT	ATN	IOP	MILOP	MSAFE	MEP
AIRCRAFT							
HU25	2960	1427	123	4	1020	15	50
HH65	2158	17069	603	74	418	47	998
HH3	4250	4891			183	16	170
C130	1723	2048			229		
SURFACE							
WPB82	11382	32073				738	357
WPB110	993	36762			6195		50
WTGB	2875	2233		2992	415	136	
WLB180	681	2121	23269	1800	123		5
WLM157		491	11508				
WLM133		1001	16450	472	770		548
WMEC270	410	34150					
WMEC210		42558			641		

Figure 11: Model Allocation of Platform Hours

PLATFORM	MISSION					
	SAR	ELT	ATN	MILOP	MSAFE	MEP
AIRCRAFT						
HU25	2221	900	79			
HH65		10320				
HH3	2296	2800			16	170
C130	123	2048		229		
SURFACE						
WPB82	5200	7848			66	86
WPB110	255	30262		1483		
WLB180		669	7331			
WLM133		231	3619			
WMEC270		27171.5				
WMEC210		38868.5				

Figure 12: Allocation of Platform Hours for Seventh District

Given that the above data set is meaningful, two types of analysis of interest to the Coast Guard were performed. The first one concerns how a change in the difficulty of the mission requirements effect the reallocation of platforms. The other concerns a similar effect due to changes in requirements of a mission type. As before, it is also assumed that there is no cross utilization between air and surface platforms. This assumption is necessary because of the fact that the Abstract of Operations did not record the cross utilization of air and surface platforms and that ZOOM would take an inordinate amount of time otherwise.

The assumption of no cross utilization allows the model to be decomposed into two smaller problems, aircraft allocation and surface platform allocation. The independently solved problems can be combined to achieve the complete solution.

The original data set was used once again to examine the effects of this decomposition. The sum of the resulting two sub-problems contained 2,721 continuous variables, 44 discrete variables, and 1,181 equations and took 72 CPU seconds to obtain a solution within eight percent of optimality. This eight percent optimality yields an optimal objective function value of $170.76 million instead of $169.07 million from the run with seven percent optimality. Figure 13 indicates the combined force structure proposed by the two sub models. This 95% reduction in computational time provided the necessary speed to complete the following two analyses.

PLATFORM	DISTRICT						
	1ST	5TH	7TH	8TH	9TH	AREA	TOTAL
AIRCRAFT							
HU25	1	0	2	0	0	-	3
HH65	7	6	16	8	2	-	39
HH3	3	1	13	0	1	-	18
C130	1	1	1	0	0	-	3
SURFACE							
WPB82	8	5	15	5	0	-	33
WPB110	2	2	16	2	0	-	22
WTGB	3	1	1	0	4	-	9
WLB180	0	2	3	0	3	-	8
WLM157	2	3	0	0	0	-	5
WLM133	1	0	1	4	0	-	6
WHEC	-	-	-	-	-	0	0
WMEC270	-	-	-	-	-	4	4
WMEC210	-	-	-	-	-	10	10

Figure 13: Models' Proposed Total Force Structure.

Changing the mission difficulty: To examine how the change in the difficulty of the missions effects the model solution, the required mission hours were modified as follows.

Ten percent of required hours from Class II mission were shifted to Class III. Similarly, ten percent from Class I were also shifted to Class II. Recall that Class I mission hours are the easiest and Class III hours the most difficult. Therefore, the effect of this shifting of hours was to make the overall hours more difficult to perform. Since some hours were removed from Class II and some hours were added to Class II, the total percentage of hours changed was 9.9 percent, slightly less than the ten percent shifted between the classes. Figure 14 provides the revised force structure proposed by the model.

PLATFORM	DISTRICT						
	1ST	5TH	7TH	8TH	9TH	AREA	TOTAL
AIRCRAFT							
HU25	1	0	2	0	1	-	4
HH65	7	6	16	8	2	-	39
HH3	4	1	13	0	0	-	18
C130	0	1	1	0	0	-	2
SURFACE							
WPB82	8	5	13	5	0	-	31
WPB110	2	2	16	2	0	-	22
WTGB	3	1	0	0	5	-	9
WLB180	1	1	0	1	2	-	5
WLM157	0	3	1	1	0	-	5
WLM133	2	0	2	2	0	-	6
WHEC	-	-	-	-	-	1	1
WMEC270	-	-	-	-	-	6	6
WMEC210	-	-	-	-	-	10	10

Figure 14: Force Structure For More Difficult Missions.

Numerous changes to the force structure were evident and are summarized below in Figure 15.

PLATFORM	DISTRICT					
	1ST	5TH	7TH	8TH	9TH	AREA
AIRCRAFT						
HU25					+1	-
HH65						-
HH3	+1				-1	-
C130	-1					-
SURFACE						
WPB82			-2			-
WPB110						-
WTGB			-1		+1	-
WLB180	+1	-1	-3	+1	-1	-
WLM157	-2		+1	+1		-
WLM133	+1		+1	-2		-
WHEC	-	-	-	-	-	+1
WMEC270	-	-	-	-	-	+2
WMEC210	-	-	-	-	-	

Figure 15: Changes in Force Structure Due to Increased Difficulty.

Interestingly, the overall number of platforms required to meet the new mission hour requirements actually decreased by two. However, the platforms not allocated were generally those designed for easier mission hours and consequently, less expensive than the more capable platforms. The number of area platforms, considered the most capable for a wide range of missions, increased by three due to the change in mission requirements. Considering that the force structure problem is concerned with the location of platforms as well as the total number of platforms, the shifting of platforms from one district to another must be considered a change in the force structure. By this definition of change, the 9.9 percent change in mission difficulty resulted in 26 changes or a 16.25

percent change in the force structure. This indicates small changes in difficulty of the missions produces significant changes in the force structure. This emphasizes the necessity of having accurate required mission hours.

Increasing required number of ELT mission hours: To examine the impact of increased operations within a particular mission. ELT was chosen for this example since its required mission hours accounted for over half of the total mission hours accomplished by the Coast Guard in Fiscal Year 1989. Figures 16 and 17 graphically illustrate changes in the force structure caused by this increase in required ELT mission hours.

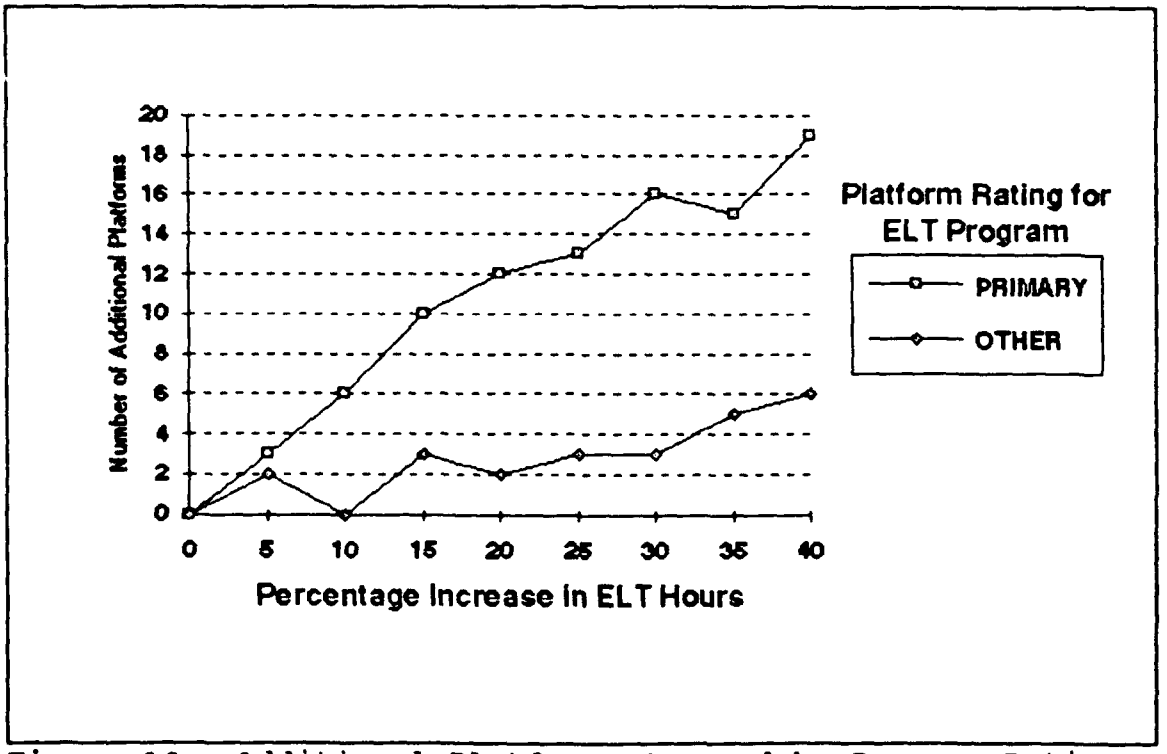

Figure 16: Additional Platforms Grouped by Program Rating

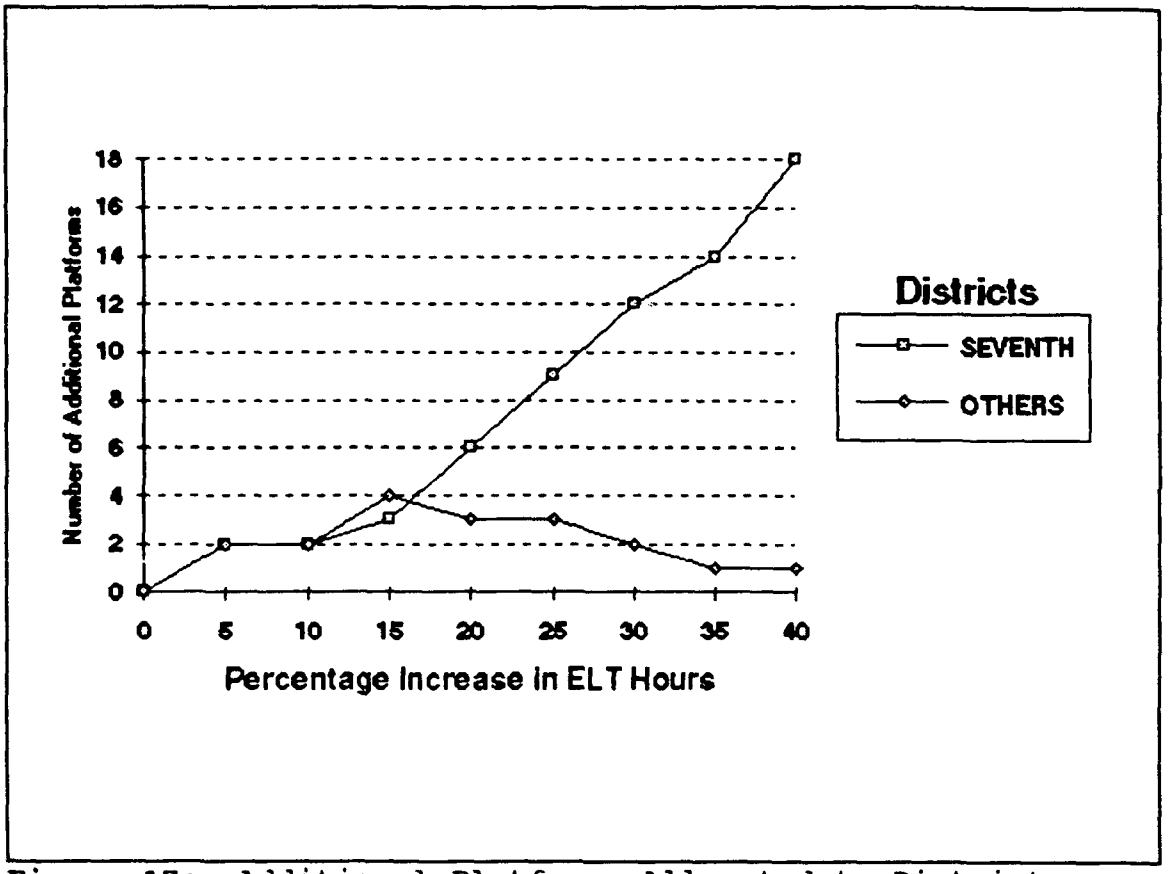

Figure 17: Additional Platforms Allocated to Districts

Recall when using the original data set of required mission hours, the model does not utilize all available platforms. As such, the additional platforms described along the y-axis of Figures 16 and 17 are not new acquisitions, rather they are platforms previously not utilized by the model. In this case, a 40 percent increase in required ELT mission hours or, equivalently, 27.2 percent increase in overall mission hours, resulted in only a 15 percent increase in the number of platforms. This increase in the number of

platforms was relatively small because of the multi-mission capability of the platforms.

Figure 16 depicts the increase in the utilization of platforms separated into two groups. The first group consists of platforms primarily designed for ELT as indicated in ROC documentation. The other consists of the remaining platforms. Clearly, this figure shows a marked increase of approximately one platform for every two percent increase in mission hours for platforms primarily designed for ELT. However, the remaining platforms increased at a slower rate, approximately one for every six percent. One conclusion for this behavior is that platforms are now being used for their primary mission which in this case is ELT.

Force structure changes in the Seventh district was of interest since originally 68 percent of the required ELT mission hours were in this district. Figure 17 compares the increase in district platforms allocated to both the Seventh district and the remaining Atlantic area districts. As expected, the number of platforms increased more rapidly in this district because it has the largest percentage of required ELT mission hours. At a 15 percent increase in required ELT hours, those area platforms previous unused by the model were being allocated to accomplish the increasing number of ELT mission hours. At this point two additional area platforms were allocated solely for accomplishing ELT hours in the Seventh district. After the 15 percent increase

one additional area platform was allocated for each five percent increase in required mission hours.

Examining figures similar to Figure 16 and 17 can provide insight into the adequacy of the available platforms in response to an unexpected increase in mission hours such as the Cuban Boat Lift of 1980 or hurricane HUGO in 1990. Other analyses are possible by examining other portions of the model's output. This section illustrates only a few of the analytical possibilities available.

VI. CONCLUSION

This thesis developed and implemented an optimization based tool to aid the Coast Guard in establishing its force structure. Since the Coast Guard has questioned previous methods of defining mission requirements and platform utilization, Chapter II proposed new methods to systematically quantify these factors. Then, Chapter III specified an integer programming model to allocate platforms to meet the requirements in the manner specified by the utilization criteria in Chapter II.

A user friendly interface was developed in Chapter IV. This interface is designed to help users maintain and update data for the problem as well as to shield them from the mathematical programming aspects of the problem. However, the interface still provides sufficient flexibility for the user to explore different but related versions of the basic force structure problem. Finally, sample outputs and analyses using the implemented system were given in Chapter V.

A. AREAS OF FUTURE STUDY

This study provides a foundation with which the Coast Guard can develop a powerful tool to solve its force structure problem. Future investigations in this direction would not

only benefit the Coast Guard in increasing its efficiency and effectiveness, but are also necessary when facing an unfavorable budgetary environment. Among many future topics, three are presented here.

(1) As discussed earlier accurate input data is necessary to improve the usefulness of this model. Toward this end, the Coast Guard is encouraged to investigate the development of models or improved techniques for forecasting costing data and mission requirements data. With improvements to the input data, the ability to develop a multi-period model becomes a logical extension to this study. A multi-period model would then be able to look at such issues as hull life, the congressionally mandated acquisition process as defined in OMB Circular Number A-109, and service life extension programs (SLEP) such as Fleet Rehabilitation and Maintenance (FRAM) and Mid-life Maintenance Availability (MMA). All of these issues could then be used to determine the life cycle cost of each platform vice the simple annual costs currently used in this model, providing a more realistic view of the actual cost of a platform type.

(2) Possible errors within the GAMS/ZOOM software were discovered during the implementation of the mathematical model and have been documented in previous studies using GAMS/ZOOM [Ref. 17]. The Coast Guard should investigate the possibility of procuring another commercial mixed integer solver for use with GAMS or the development of a customized problem generator

and integer solver written in a general-purpose compiled language (e.g., FORTRAN).

(3) The research level front end interface is admittedly rudimentary and was developed to provide an example of the ability to insulate the user from an underlying mathematical model. An improved interface would greatly enhance the usability and acceptance of the optimization model. The Coast Guard is presently investigating a KSS (knowledge-based decision support systems) project using a DSS (decision support system) environment [Ref. 6]. The "DSS is an interactive software tool for working with models and data" [Ref. 6:p. 6]. The combination of this mathematical model with an interface based on the KSS environment would provide the Coast Guard with a necessary tool for analyzing the many force structure issues.

In closing, it should be mentioned that this study provides the Coast Guard with one method of addressing its force structure problem. Since no one, mathematical or otherwise, model can completely capture every facet of the problem, the Coast Guard is also encouraged to consider other approaches, e.g., stochastic or statistical modeling, along with directions mentioned above.

APPENDIX A

INTERFACE OUTPUT: GAMS DATA SET

```
SETS

 P   ALL PLATFORMS
 /
   HU25     ,HH65     ,HH60     ,C130     ,WPB82    ,WPB110   ,
   WTGB     ,WLB180   ,WLM157   ,WLM133   ,WHEC     ,WMEC270  ,
   WMEC210 /

  IDS(P)  DISTRICT SURFACE PLATFORMS
 /
     WPB82    ,
     WPB110   ,
     WTGB     ,
     WLB180   ,
     WLM157   ,
     WLM133     /

  IDA(P)  DISTRICT AIR     PLATFORMS
 /
     HU25     ,
     HH65     ,
     HH60     ,
     C130       /

  IAS(P)  AREA SURFACE     PLATFORMS
 /
     WHEC     ,
     WMEC270  ,
     WMEC210    /

 D  DISTRICT
 /
   FIRST      ,FIFTH      ,SEVENTH    ,EIGHTH     ,
   NINTH      /

 M  MISSION
 / SAR   ,ELT   ,ATN   ,IOP   ,MILOP,MSAFE,MER  /

 L  CLASS                  /1, 2, 3/;

 SET INDP(D)  INDEPENDENT DISTRICTS     /  NINTH /;
```

```
   ALIAS(D,DP);
TABLE
   ADS(IDS,M)  MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IDS)
*              CAN PERFORM MISSION (M).
              SAR   ELT   ATN   IOP   MILOP MSAFE MER
   WPB82      1.00  1.00  0.00  0.00  0.00  0.10  0.10
   WPB110     0.50  1.00  0.00  0.00  0.50  0.50  0.10
   WTGB       1.00  0.50  0.00  1.00  0.10  0.50  0.00
   WLB180     0.50  0.10  1.00  0.10  0.10  0.10  0.10
   WLM157     0.50  0.10  1.00  0.10  0.10  0.00  0.10
   WLM133     0.50  0.10  1.00  0.10  0.10  0.00  0.10
                                                      ;

TABLE
   BDS(IDS,L)  MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IDS)
*              CAN PERFORM AT CLASS (L).
                 1       2       3
   WPB82         1.00    0.10    0.00
   WPB110        1.00    0.50    0.10
   WTGB          1.00    0.50    0.10
   WLB180        1.00    1.00    0.50
   WLM157        1.00    1.00    0.10
   WLM133        1.00    0.50    0.10
                                      ;

TABLE
   ADA(IDA,M)  MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IDA)
*              CAN PERFORM MISSION (M).
              SAR   ELT   ATN   IOP   MILOP MSAFE MER
   HU25       1.00  1.00  0.10  0.10  0.50  0.10  0.50
   HH65       1.00  1.00  0.50  0.50  0.50  0.10  0.50
   HH60       1.00  1.00  0.50  0.00  0.50  0.10  0.50
   C130       1.00  1.00  0.50  0.50  0.50  0.50  0.50
                                                      ;

TABLE
   BDA(IDA,L)  MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IDA)
*              CAN PERFORM AT CLASS (L).
                 1       2       3
   HU25          1.00    1.00    0.10
   HH65          1.00    0.50    0.00
   HH60          1.00    1.00    0.50
   C130          1.00    1.00    1.00
                                      ;

TABLE
   AAS(IAS,M)  MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IAS)
*              CAN PERFORM MISSION (M).
              SAR   ELT   ATN   IOP   MILOP MSAFE MER
   WHEC       0.50  1.00  0.00  0.00  1.00  0.00  0.10
   WMEC270    0.50  1.00  0.00  0.00  1.00  0.50  0.00
   WMEC210    0.50  1.00  0.00  0.00  0.50  0.50  0.10
```

;

TABLE
BAS(IAS,L) MAXIMUM PERCENTAGE OF TIME THAT PLATFORM (IAS)
* CAN PERFORM AT CLASS (L).

	1	2	3
WHEC	1.00	1.00	1.00
WMEC270	1.00	1.00	0.50
WMEC210	1.00	1.00	0.50

;

TABLE
HRSURF(D,M,L) SURFACE RESOURCE HOURS REQUIRED FOR MISSION
* (M) IN DISTRICT (D) AT CLASS (L) PER YEAR.

		1	2	3
FIRST	.SAR	5000	1000	65
FIRST	.ELT	8000	6000	1416
FIRST	.ATN	5000	3000	1125
FIRST	.IOP	800	434	200
FIRST	.MILOP	1500	1000	250
FIRST	.MSAFE	500	50	0
FIRST	.MER	500	48	0
FIFTH	.SAR	1000	1200	61
FIFTH	.ELT	7000	4800	1045
FIFTH	.ATN	6000	4000	1314
FIFTH	.IOP	0	0	0
FIFTH	.MILOP	1100	1065	900
FIFTH	.MSAFE	59	0	0
FIFTH	.MER	200	71	0
SEVENTH	.SAR	4000	1200	255
SEVENTH	.ELT	40000	35050	30000
SEVENTH	.ATN	10000	900	50
SEVENTH	.IOP	0	0	0
SEVENTH	.MILOP	1000	283	200
SEVENTH	.MSAFE	66	0	0
SEVENTH	.MER	86	0	0
EIGHTH	.SAR	1800	620	0
EIGHTH	.ELT	6090	7700	4000
EIGHTH	.ATN	12700	1070	0
EIGHTH	.IOP	0	0	0
EIGHTH	.MILOP	512	150	150
EIGHTH	.MSAFE	63	0	0
EIGHTH	.MER	50	0	0
NINTH	.SAR	140	0	0
NINTH	.ELT	289	0	0
NINTH	.ATN	2300	2000	1769
NINTH	.IOP	600	2000	1230
NINTH	.MILOP	35	0	0
NINTH	.MSAFE	136	0	0
NINTH	.MER	5	0	0

;

TABLE
HRAIR(D,M,L) AIR RESOURCE HOURS REQUIRED FOR MISSION (M)
* IN DISTRICT (D) AT CLASS (L) PER YEAR.

		1	2	3
FIRST	.SAR	668	800	400
FIRST	.ELT	800	1900	1336
FIRST	.ATN	200	44	0
FIRST	.IOP	0	0	4
FIRST	.MILOP	600	300	233
FIRST	.MSAFE	10	5	0
FIRST	.MER	278	200	50
FIFTH	.SAR	900	600	360
FIFTH	.ELT	1070	700	500
FIFTH	.ATN	100	71	0
FIFTH	.IOP	0	1	0
FIFTH	.MILOP	121	100	70
FIFTH	.MSAFE	18	0	0
FIFTH	.MER	149	100	0
SEVENTH	.SAR	2300	1300	1040
SEVENTH	.ELT	5600	5300	5168
SEVENTH	.ATN	79	0	0
SEVENTH	.IOP	0	0	0
SEVENTH	.MILOP	60	69	100
SEVENTH	.MSAFE	10	6	0
SEVENTH	.MER	120	50	0
EIGHTH	.SAR	800	870	0
EIGHTH	.ELT	1600	1420	0
EIGHTH	.ATN	63	0	0
EIGHTH	.IOP	0	0	0
EIGHTH	.MILOP	100	61	0
EIGHTH	.MSAFE	10	4	0
EIGHTH	.MER	106	100	0
NINTH	.SAR	700	354	0
NINTH	.ELT	42	0	0
NINTH	.ATN	150	19	0
NINTH	.IOP	50	23	0
NINTH	.MILOP	26	10	0
NINTH	.MSAFE	12	3	0
NINTH	.MER	55	0	0

;

TABLE
CURRENT(P,D) CURRENT FORCE STRUCTURE FOR THE COAST GUARD

	FIRST	FIFTH	SEVENTH	EIGHTH	NINTH
HU25	5	0	10	3	0
HH65	5	3	16	11	4
HH60	4	3	9	0	2
C130	0	5	8	0	0
WPB82	8	8	8	10	0
WPB110	2	2	16	2	0
WTGB	3	1	0	0	5
WLB180	3	3	2	2	5
WLM157	2	3	0	0	0

```
WLM133      3          0          1          2          0
WHEC        1          0          0          0          0
WMEC270     2          5          3          0          0
WMEC210     1          3          4          2          0
                                                        ;

SCALARS

  PICK       FLAG FOR WHICH OBJECTIVE FUNCTION TO USE
  CNT        PLATFORM COUNT
  ACQLIM     FLAG FOR LIMITING ACQUISITION COSTS
  MOVLIM     FLAG FOR LIMITING MOVEMENT COSTS
  LIMACQ     LIMIT FOR ACQUISITION COSTS
  LIMMOV     LIMIT FOR MOVEMENT COSTS
  PENALTY1 ACQUISITION COST CONSTANT
  PENALTY2 MOVEMENT COST CONSTANT

  NOIDS      FLAG FOR NO DISTRICT SURFACE PLATFORMS DEFINED
  NOIAS      FLAG FOR NO AREA SURFACE PLATFORMS DEFINED
  NOIDA      FLAG FOR NO DISTRICT AIR PLATFORMS DEFINED


  PARAMETERS

     AIRPCT(M)   % OF SURFACE RESOURCE HOURS AIRCRAFT CAN
                   ACCOMPLISH
        /  SAR      0.00
           ELT      0.00
           ATN      0.00
           IOP      0.00
           MILOP    0.00
           MSAFE    0.00
           MER      0.00  /

     SURPCT(M)   % OF AIR RESOURCE HOURS CUTTERS CAN ACCOMPLISH
        /  SAR      0.00
           ELT      0.00
           ATN      0.00
           IOP      0.00
           MILOP    0.00
           MSAFE    0.00
           MER      0.00  /


     RHDS(IDS)  MAXIMUM NUMBER OF RESOURCE HOURS PLATFORM (IDS)
*                 CAN PERFORM PER YEAR.
                /   WPB82          1650
                    WPB110         2000
                    WTGB           2300
                    WLB180         4000
                    WLM157         4000
```

```
                WLM133          3850   /

    RHDA(IDA)  MAXIMUM NUMBER OF RESOURCE HOURS PLATFORM (IDA)
*                CAN PERFORM PER YEAR.
             /   HU25             800
                 HH65             645
                 HH60             700
                 C130             800   /

    RHAS(IAS)  MAXIMUM NUMBER OF RESOURCE HOURS PLATFORM (IAS)
*                CAN PERFORM PER YEAR.
             /   WHEC            4320
                 WMEC270         4320
                 WMEC210         4320   /


    COSTDS(IDS)  OPERATING COSTS FOR PLATFORM (IDS) PER YEAR IN
*                MILLIONS OF DOLLARS.
             /   WPB82          0.370
                 WPB110         0.590
                 WTGB           0.650
                 WLB180         1.580
                 WLM157         1.130
                 WLM133         0.990   /

    COSTDA(IDA)  OPERATING COSTS FOR PLATFORM (IDA) PER YEAR IN
*                MILLIONS OF DOLLARS.
             /   HU25           1.640
                 HH65           1.110
                 HH60           1.470
                 C130           2.220   /

    COSTAS(IAS)  OPERATING COSTS FOR PLATFORM (IAS) PER YEAR IN
*                MILLIONS OF DOLLARS.
             /   WHEC           4.280
                 WMEC270        2.690
                 WMEC210        2.130   /

  TOTCURRENT(P)   TOTAL AVAILABLE PLATFORMS

  ULDS(IDS,M,L)   UPPER UTILIZATION LIMITS FOR RESOURCE HOURS
                  EXPENDED ON MISSION (M) AT CLASS (L)
                  FOR DISTRICT SURFACE PLATFORMS (IDS)
  ULAS(IAS,M,L)   UPPER UTILIZATION LIMITS FOR RESOURCE HOURS
                  EXPENDED ON MISSION (M) AT CLASS (L)
                  FOR AREA SURFACE PLATFORMS (IAS)
  ULDA(IDA,M,L)   UPPER UTILIZATION LIMITS FOR RESOURCE HOURS
                  EXPENDED ON MISSION (M) AT CLASS (L)
                  FOR DISTRICT AIR PLATFORMS (IDA);


    ULDS(IDS,M,L) = ADS(IDS,M)*BDS(IDS,L);
```

```
ULAS(IAS,M,L) = AAS(IAS,M)*BAS(IAS,L);
ULDA(IDA,M,L) = ADA(IDA,M)*BDA(IDA,L);


RHDS(IDS) = RHDS(IDS)/100.0;
RHDA(IDA) = RHDA(IDA)/100.0;
RHAS(IAS) = RHAS(IAS)/100.0;
HRSURF(D,M,L) = HRSURF(D,M,L)/100.0;
HRAIR(D,M,L) = HRAIR(D,M,L)/100.0;
TOTCURRENT(P) = SUM(D,CURRENT(P,D));

NOIDS = 1;
NOIDA = 1;
NOIAS = 1;

PICK = 1.0;
ACQLIM =   1.0000;
MOVLIM =   1.0000;
LIMACQ =   0.0000;
LIMMOV =   0.0000;
PENALTY1 =   0.5000;
PENALTY2 =   0.1000;
```

APPENDIX B

GAMS FORMULATION

```
$TITLE       United States Coast Guard Fleet-Mix Problem
$STITLE      LT J.E. TOMKO, USCG

*-------GAMS and Dollar Control Options----------------------

$OFFUPPER OFFSYMXREF OFFSYMLIST

OPTIONS LIMCOL = 0, LIMROW = 0, SOLPRINT = OFF;
OPTIONS RESLIM = 50000, ITERLIM = 2500000;
OPTIONS WORK = 35000;

*-------Definitions and Data---------------------------------

OPTIONS OPTCR = 0.20;

$INCLUDE GAMS DATA A

*----------------Model-----------------------------------------
  VARIABLES
     TCOST          total operating cost of fleet per year;

  INTEGER VARIABLES
     NXDS(IDS,D)     new # of district surface platforms (IDS)
                     in district (D)
     NXAS(IAS)       new # of area surface platforms (IAS)
     NXDA(IDA,D)     new # of district air platforms (IDA)
                     in district (D)
     OXDS(IDS,D)     current # of district surface platforms (IDS)
                     in district (D)
     OXAS(IAS)       current # of area surface platforms (IAS)
     OXDA(IDA,D)     current # of district air platforms (IDA)
                     in district (D);

       OXDS.UP(IDS,D) = TOTCURRENT(IDS);
       OXDA.UP(IDA,D) = TOTCURRENT(IDA);
       OXAS.UP(IAS)   = TOTCURRENT(IAS);

  POSITIVE VARIABLES
    YDS(IDS,D,M,L) # hrs of platform (IDS) in district (D) assigned
                     to mission area (M) at class (L) performed
                     for surface missions
    YAS(IAS,D,M,L) # hrs of platform (IAS) in district (D) assigned
                     to mission area (M) at class (L) performed
                     for surface missions
```

```
  YDA(IDA,D,M,L) # hrs of platform (IDA) in district (D) assigned
                   to mission area (M) at classs (L) performed
                   for air missions
  MS(IDS,D,DP)   movement variable for platform type (IDS)
  MA(IDA,D,DP)   movement variable for platform type (IDA)
  ZDS(IDS,D,M,L) # hrs of platform (IDS) in district (D) assigned
                   to mission area (M) at class (L) performed
                   for air missions
  ZAS(IAS,D,M,L) # hrs of platform (IAS) in district (D) assigned
                   to mission area (M) at class (L) performed
                   for air missions
  ZDA(IDA,D,M,L) # hrs of platform (IDA) in district (D) assigned
                   to mission area (M) at class (L) performed
                   for surface missions


EQUATIONS
    OBJ1                 total operating cost
    OBJ2                 operating cost without acquisition costs
    OBJ3                 operating cost without movement costs
    AVAILAS(IAS)         observe current force structure of IAS
    AIRHRREQ(D,M,L)      meet all AIR resource hour requirements
    SURHRREQ(D,M,L)          meet all SURFACE resource hour
                         requirements
    PCTAIR(D,M)          satistfy AIR TO SURFACE MAX PERCENTAGE
    PCTSUR(D,M)          satistfy SURFACE TO AIR MAX PERCENTAGE
    DISSURF(IDS,D)      Produce enough district surface platforms
    DISAIR(IDA,D)        Produce enough district air platforms
    AREA(IAS)            Produce enough area surface platforms
    USEDSUR(IDS,D,M,L)  Satisfy max usage of district surface
                         units
    USEASUR(IAS,M,L)     Satisfy max usage of area surface units
    USEDAIR(IDA,D,M,L)  Satisfy max usage of district air units
    ACQ                  Limit total acquisition costs
    MOV                  Limit total movement costs
    SUPXDS(IDS,D)        Do not exceed supply of ids platforms
    DEMXDS(IDS,DP)       Meet demand for ids platforms
    SUPXDA(IDA,D)        Do not exceed supply of ida platforms
    DEMXDA(IDA,DP)       Meet demand for ida platforms;

* minimize
    OBJ1..
        TCOST =E= SUM(IAS, COSTAS(IAS)*(OXAS(IAS) + NXAS(IAS)))
            +SUM((IDS,D),COSTDS(IDS)*(OXDS(IDS,D)+NXDS(IDS,D)))
            +SUM((IDA,D),COSTDA(IDA)*(OXDA(IDA,D)+NXDA(IDA,D)))
            + SUM(IAS, PENALTY1*NXAS(IAS))
            + SUM((IDS,D), PENALTY1*NXDS(IDS,D))
            + SUM((IDA,D), PENALTY1*NXDA(IDA,D))
            + PENALTY2 * SUM((IDA,D,DP)$(ORD(D) NE ORD(DP)),
                             MA(IDA,D,DP))
            + PENALTY2 * SUM((IDS,D,DP)$(ORD(D) NE ORD(DP)),
                             MS(IDS,D,DP));
```

```
      OBJ2$(pick eq 2)..
          TCOST =E= SUM(IAS, COSTAS(IAS)*(OXAS(IAS) + NXAS(IAS)))
              +SUM((IDS,D),COSTDS(IDS)*(OXDS(IDS,D)+NXDS(IDS,D)))
              +SUM((IDA,D),COSTDA(IDA)*(OXDA(IDA,D)+NXDA(IDA,D)))
              + SUM(IAS, PENALTY1*NXAS(IAS))
              + SUM((IDS,D), PENALTY1*NXDS(IDS,D))
              + SUM((IDA,D), PENALTY1*NXDA(IDA,D));

      OBJ3$(pick eq 3)..
           TCOST =E= SUM(IAS, COSTAS(IAS)*(OXAS(IAS) + NXAS(IAS)))
              +SUM((IDS,D),COSTDS(IDS)*(OXDS(IDS,D)+NXDS(IDS,D)))
              +SUM((IDA,D),COSTDA(IDA)*(OXDA(IDA,D)+NXDA(IDA,D)))
              + PENALTY2 * SUM((IDA,D,DP)$(ORD(D) NE ORD(DP)),
                               MA(IDA,D,DP))
              + PENALTY2 * SUM((IDS,D,DP)$(ORD(D) NE ORD(DP)),
                               MS(IDS,D,DP));

* subject to
      AVAILAS(IAS)..   OXAS(IAS)   =L= SUM(D, CURRENT(IAS,D));

      AIRHRREQ(D,M,L)$(HRAIR(D,M,L) GT 0)..
                    SUM(IDA$(ULDA(IDA,M,L) GT 0),
                        YDA(IDA,D,M,L)) +
                    SUM(IDS$(ULDS(IDS,M,L) GT 0 AND SURPCT(M) GT 0),
                        ZDS(IDS,D,M,L)) +
                    SUM(IAS$(ULAS(IAS,M,L) GT 0 AND SURPCT(M) GT 0),
                        ZAS(IAS,D,M,L)) =G= HRAIR(D,M,L);

      SURHRREQ(D,M,L)$(HRSURF(D,M,L) GT 0)..
                    SUM(IDA$(ULDA(IDA,M,L) GT 0 AND AIRPCT(M) GT 0),
                        ZDA(IDA,D,M,L)) +
                    SUM(IDS$(ULDS(IDS,M,L) GT 0),
                        YDS(IDS,D,M,L)) +
                    SUM(IAS$(ULAS(IAS,M,L) GT 0),
                        YAS(IAS,D,M,L)) =G= HRSURF(D,M,L);

      PCTAIR(D,M)$(AIRPCT(M) GT 0)..
                   SUM((L,IDA)$(ULDA(IDA,M,L) GT 0),
                        ZDA(IDA,D,M,L)) =L=
                   AIRPCT(M) * SUM(L, HRSURF(D,M,L));

      PCTSUR(D,M)$(SURPCT(M) GT 0)..
                    SUM((IAS,L)$(ULAS(IAS,M,L) GT 0),
                         ZAS(IAS,D,M,L)) +
                    SUM((IDS,L)$(ULDS(IDS,M,L) GT 0),
                         ZDS(IDS,D,M,L)) =L=
                    SURPCT(M) * SUM(L, HRAIR(D,M,L));

      DISSURF(IDS,D)..
                    SUM((M,L)$(ULDS(IDS,M,L) GT 0),
                        YDS(IDS,D,M,L) + ZDS(IDS,D,M,L))
```

```
                =L= RHDS(IDS) * (OXDS(IDS,D)+NXDS(IDS,D));

DISAIR(IDA,D)..
                SUM((M,L)$(ULDA(IDA,M,L) GT 0),
                    YDA(IDA,D,M,L) + ZDA(IDA,D,M,L))
                 =L= RHDA(IDA) * (OXDA(IDA,D)+NXDA(IDA,D));
AREA(IAS)..
                 SUM((D,M,L)$(ULAS(IAS,M,L) GT 0),
                       YAS(IAS,D,M,L) + ZAS(IAS,D,M,L))
                  =L=  RHAS(IAS) * (OXAS(IAS)+NXAS(IAS));

USEDSUR(IDS,D,M,L)$(ULDS(IDS,M,L) GT 0)..
                   YDS(IDS,D,M,L) + ZDS(IDS,D,M,L)
                     =L= (OXDS(IDS,D)+NXDS(IDS,D))
                          * RHDS(IDS) * ULDS(IDS,M,L);

USEASUR(IAS,M,L)$(ULAS(IAS,M,L) GT 0)..
                 SUM(D, YAS(IAS,D,M,L) + ZAS(IAS,D,M,L))
                   =L= RHAS(IAS) * ULAS(IAS,M,L) *
                       (OXAS(IAS)+NXAS(IAS));
USEDAIR(IDA,D,M,L)$(ULDA(IDA,M,L) GT 0)..
                 YDA(IDA,D,M,L) + ZDA(IDA,D,M,L)
                   =L= RHDA(IDA) * (OXDA(IDA,D)+NXDA(IDA,D))
                       * ULDA(IDA,M,L);

SUPXDS(IDS,D)..     SUM(DP, MS(IDS,D,DP)) =L= CURRENT(IDS,D);
DEMXDS(IDS,DP)..    SUM(D, MS(IDS,D,DP)) =E= OXDS(IDS,DP);
SUPXDA(IDA,D)..     SUM(DP, MA(IDA,D,DP)) =L= CURRENT(IDA,D);
DEMXDA(IDA,DP)..    SUM(D, MA(IDA,D,DP)) =E= OXDA(IDA,DP);

ACQ$(ACQLIM EQ 1)..
          PENALTY1 * (SUM((D,IDA), NXDA(IDA,D)) +
                      SUM((D,IDS), NXDS(IDS,D)) +
                      SUM(IAS, NXAS(IAS))) =L= LIMACQ;

MOV$(MOVLIM EQ 1)..
           PENALTY2 * SUM((IDA,D,DP)$(ORD(D) NE ORD(DP)),
                          MA(IDA,D,DP)) +
           PENALTY2 * SUM((IDS,D,DP)$(ORD(D) NE ORD(DP)),
                          MS(IDS,D,DP)) =L= LIMMOV;

   YAS.FX(IAS,INDP,M,L)$(CARD(INDP) GT 0) = 0.0;
   YAS.FX(IAS,D,M,L)$(NOIAS EQ 0) = 0.0;
   YDS.FX(IDS,D,M,L)$(NOIDS EQ 0) = 0.0;
   YDA.FX(IDA,D,M,L)$(NOIDA EQ 0) = 0.0;
   YAS.FX(IAS,D,M,L)$(ULAS(IAS,M,L) EQ 0) = 0.0;
   YDS.FX(IDS,D,M,L)$(ULDS(IDS,M,L) EQ 0) = 0.0;
   YDA.FX(IDA,D,M,L)$(ULDA(IDA,M,L) EQ 0) = 0.0;
   ZDA.FX(IDA,D,M,L)$(ULDA(IDA,M,L) EQ 0) = 0.0;
   ZAS.FX(IAS,D,M,L)$(ULAS(IAS,M,L) EQ 0) = 0.0;
   ZDS.FX(IDS,D,M,L)$(ULDS(IDS,M,L) EQ 0) = 0.0;
   ZDA.FX(IDA,D,M,L)$(AIRPCT(M) EQ 0) = 0.0;
```

```
      ZAS.FX(IAS,D,M,L)$(SURPCT(M) EQ 0) = 0.0;
      ZDS.FX(IDS,D,M,L)$(SURPCT(M) EQ 0) = 0.0;

  MODEL FLEETMIX /ALL/;
  SOLVE FLEETMIX USING RMIP MINIMIZING TCOST;
      CNT    = SUM((IDS,D), OXDS.L(IDS,D) + NXDS.L(IDS,D)) +
               SUM((IDA,D), OXDA.L(IDA,D) + NXDA.L(IDA,D)) +
               SUM(IAS, OXAS.L(IAS) + NXAS.L(IAS));
  EQUATIONS
      TOTAL   minimum total number of platforms needed;

      TOTAL.. SUM((IDS,D), OXDS(IDS,D) + NXDS(IDS,D)) +
               SUM((IDA,D), OXDA(IDA,D) + NXDA(IDA,D)) +
               SUM(IAS, OXAS(IAS) + NXAS(IAS)) =G= FLOOR(CNT);
  MODEL FLEET /ALL/
  SOLVE FLEET USING MIP MINIMIZING TCOST;

*-------------------Reports-----------------------------------
* print the optimal objective value and solution

  OPTION DECIMALS = 5;
  DISPLAY TCOST.L,CNT;

  PARAMETER COST(*,*);

  COST(IDS,D)                      =  OXDS.L(IDS,D);
  COST(IDS,'TYPE TOTAL')           =  SUM(D, OXDS.L(IDS,D));
  COST(IDS,'UNIT COST')            =  COSTDS(IDS) * 1000000;
  COST(IDS,'TOTAL COST')           =  COSTDS(IDS) * 1000000 *
                                      SUM(D, OXDS.L(IDS,D));
  COST(IDA,D)                      =  OXDA.L(IDA,D);
  COST(IDA,'TYPE TOTAL')           =  SUM(D, OXDA.L(IDA,D));
  COST(IDA,'UNIT COST')            =  COSTDA(IDA) * 1000000;
  COST(IDA,'TOTAL COST')           =  COSTDA(IDA) * 1000000 *
                                      SUM(D, OXDA.L(IDA,D));
  COST(IAS,'TYPE TOTAL')           =  OXAS.L(IAS);
  COST(IAS,'UNIT COST')            =  COSTAS(IAS) * 1000000;
  COST(IAS,'TOTAL COST')           =  COSTAS(IAS)*1000000*
                                                OXAS.L(IAS);
  COST('TOTAL','TOTAL COST')          =   SUM(IDS,  COST(IDS,'TOTAL
                                          COST')) +
                                      SUM(IDA,   COST(IDA,'TOTAL
                                          COST')) +
                                      SUM(IAS,   COST(IAS,'TOTAL
                                          COST'));
  OPTION DECIMALS = 2;
  DISPLAY COST;

  PARAMETER SHORTFALL(*,*) ACQUISITIONS RECOMMENDED;

  SHORTFALL(IDS,D)                 =  NXDS.L(IDS,D);
  SHORTFALL(IDS,'TYPE TOTAL')      =  SUM(D, NXDS.L(IDS,D));
```

```
SHORTFALL(IDS,'UNIT COST')       =  COSTDS(IDS)*1000000;
SHORTFALL(IDS,'ACQ COST')        =  PENALTY1*1000000;
SHORTFALL(IDS,'TOTAL COST')     =  (COSTDS(IDS)+PENALTY1)*1000000
                                    * SUM(D, NXDS.L(IDS,D));
SHORTFALL(IDA,D)                 =  NXDA.L(IDA,D);
SHORTFALL(IDA,'TYPE TOTAL')      =  SUM(D, NXDA.L(IDA,D));
SHORTFALL(IDA,'UNIT COST')       =  COSTDA(IDA)*1000000;
SHORTFALL(IDA,'ACQ COST')        =  PENALTY1*1000000;
SHORTFALL(IDA,'TOTAL COST')     =  (COSTDA(IDA)+PENALTY1)*1000000
                                    * SUM(D, NXDA.L(IDA,D));
SHORTFALL(IAS,'TYPE TOTAL')      =  NXAS.L(IAS);
SHORTFALL(IAS,'UNIT COST')       =  COSTAS(IAS)*1000000;
SHORTFALL(IAS,'ACQ COST')        =  PENALTY1*1000000;
SHORTFALL(IAS,'TOTAL COST')     =  (COSTAS(IAS)+PENALTY1)*1000000
                                    * NXAS.L(IAS);
SHORTFALL('TOTAL','TOTAL COST') =  SUM(IDS,SHORTFALL(IDS,'TOTAL
                                       COST'))
                                  + SUM(IDA,SHORTFALL(IDA,'TOTAL
                                       COST'))
                                  + SUM(IAS,SHORTFALL(IAS,'TOTAL
                                       COST'));

OPTION DECIMALS = 0;
DISPLAY $(SHORTFALL('TOTAL','TOTAL COST') GT 0) SHORTFALL;

PARAMETER MOVEMENT(*,D,DP)   PLATFORM MOVEMENT FROM (ROW) TO
                            (COLUMN);

MOVEMENT(IDS,D,DP)  =  MS.L(IDS,D,DP);
MOVEMENT(IDA,D,DP)  =  MA.L(IDA,D,DP);
OPTION MOVEMENT:2:1:1;
DISPLAY MOVEMENT;


PARAMETER ALLOCATE(D,P,M), TALL(*,M);

ALLOCATE(D,IDS,M) = SUM(L,ZDS.L(IDS,D,M,L)+YDS.L(IDS,D,M,L))
                       * 100.0;
ALLOCATE(D,IDA,M) = SUM(L,ZDA.L(IDA,D,M,L)+YDA.L(IDA,D,M,L))
                       * 100.0;
ALLOCATE(D,IAS,M) = SUM(L,ZAS.L(IAS,D,M,L)+YAS.L(IAS,D,M,L))
                       * 100.0;
OPTION ALLOCATE:2:1:1; DISPLAY ALLOCATE;

TALL(P,M)          = SUM(D, ALLOCATE(D,P,M) );
TALL('Total',M)    = SUM( (D,P), ALLOCATE(D,P,M) );
DISPLAY TALL;
```

APPENDIX C

GAMS OUTPUT

```
---- 526 VARIABLE  TCOST.L  = 172.11000 TOTAL OPERATING COST OF FLEET PER YEAR
        PARAMETER CNT      = 147.91996 PLATFORM COUNT
```

---- 547 PARAMETER COST

	FIRST	FIFTH	SEVENTH	EIGHTH	NINTH	TYPE TOTAL	UNIT COST	TOTAL COST
HU25	3.00		4.00			7.00	1640000.00	11480000.00
HH65	5.00	3.00	16.00	8.00	2.00	34.00	1110000.00	37740000.00
HH3	4.00	2.00	8.00		1.00	15.00	1470000.00	22050000.00
C130		2.00	3.00			5.00	2220000.00	11100000.00
WPB82	8.00	6.00	8.00	5.00		27.00	370000.00	9990000.00
WPB110	2.00	2.00	16.00	2.00		22.00	590000.00	12980000.00
WTGB	3.00				4.00	7.00	650000.00	4550000.00
WLB180		1.00	2.00	2.00	2.00	7.00	1580000.00	11060000.00
WLM157	1.00	2.00				3.00	1130000.00	3390000.00
WLM133	2.00			1.00	2.00	5.00	990000.00	4950000.00
WHEC							4280000.00	
WMEC270						8.00	2690000.00	21520000.00
WMEC210						10.00	2130000.00	21300000.00
TOTAL								1.721100E+8

---- 583 PARAMETER MOVEMENT PLATFORM MOVEMENT FROM (ROW) TO (COLUMN)

INDEX 1 = HU25

	FIRST	SEVENTH
FIRST	3.00	
SEVENTH		4.00

INDEX 1 = HH65

	FIRST	FIFTH	SEVENTH	EIGHTH	NINTH
FIRST	5.00				
FIFTH		3.00			
SEVENTH			16.00		
EIGHTH				8.00	
NINTH					2.00

583 PARAMETER MOVEMENT PLATFORM MOVEMENT FROM (ROW) TO (COLUMN)

INDEX 1 - HH3

	FIRST	FIFTH	SEVENTH	NINTH
FIRST	4.00			
FIFTH		2.00		
SEVENTH			8.00	
NINTH				1.00

INDEX 1 - C130

	FIFTH	SEVENTH
FIFTH	2.00	
SEVENTH		3.00

INDEX 1 - WPB82

	FIRST	FIFTH	SEVENTH	EIGHTH
FIRST	8.00			
FIFTH		6.00		
SEVENTH			8.00	
EIGHTH				5.00

INDEX 1 - WPB110

	FIRST	FIFTH	SEVENTH	EIGHTH
FIRST	2.00			
FIFTH		2.00		
SEVENTH			16.00	
EIGHTH				2.00

INDEX 1 - WTGB

	FIRST	NINTH
FIRST	3.00	
NINTH		4.00

INDEX 1 - WLB180

	FIFTH	SEVENTH	EIGHTH	NINTH
FIFTH	1.00			
SEVENTH		2.00		
EIGHTH			2.00	
NINTH				2.00

583 PARAMETER MOVEMENT PLATFORM MOVEMENT FROM (ROW) TO (COLUMN)

INDEX 1 - WLM157

	FIRST	FIFTH
FIRST	1.00	
FIFTH		2.00

INDEX 1 - WLM133

	FIRST	SEVENTH	EIGHTH
FIRST	2.00		
SEVENTH		1.00	
EIGHTH			2.00

---- 591 PARAMETER ALLOCATE

INDEX 1 - FIRST

	SAR	ELT	ATN	IOP	MILOP	MSAFE	MEP
HU25	939.50	287.50	200.00	4.00			528.00
HH65	768.50	2412.50	44.00				
HH3	160.00	1336.00				1133.00	15.00
WPB82	3330.00	9320.00					550.00
WPB110	705.00	2400.00				895.00	
WTGB	2030.00	2070.00		953.00	1085.00		
WLM157		40.00	3560.00				
WLM133		336.00	5565.00	481.00	770.00		548.00
WMEC270		850.00					

INDEX 1 - FIFTH

	SAR	ELT	ATN	IOP	MILOP	MSAFE	MEP
HH65		1275.00	171.00	1.00	221.00	18.00	249.00
HH3	260.00	995.00			70.00		
C130	1600.00						
WPB82	1580.00	7990.00				59.00	271.00
WPB110		2400.00			1600.00		
WLB180			3876.50		123.50		
WLM157		562.50	7437.50				
WMEC270	681.00	1892.50			700.00		
WMEC210					641.50		

---- 591 PARAMETER ALLOCATE

INDEX 1 = SEVENTH

	SAR	ELT	ATN	MILOP	MSAFE	MEP
HU25	2221.00	900.00	79.00			
HH65		10320.00				
HH3	2296.00	2800.00			16.00	170.00
C130	123.00	2048.00		229.00		
WPB82	5200.00	7848.00			66.00	86.00
WPB110	255.00	30262.00		1483.00		
WLB180		669.00	7331.00			
WLM133		231.00	3619.00			
WMEC270		27171.50				
WMEC210		38868.50				

INDEX 1 = EIGHTH

	SAR	ELT	ATN	MILOP	MSAFE	MEP
HH65	1670.00	3020.00	63.00	161.00	14.00	206.00
WPB82	2420.00	5830.00				
WPB110		3438.00		512.00		50.00
WLB180		1058.00	6579.00	300.00	63.00	
WLM133		509.00	7191.00			
WMEC270		3265.00				
WMEC210		3690.00				

INDEX 1 = NINTH

	SAR	ELT	ATN	IOP	MILOP	MSAFE	MEP
HH65	354.00	42.00	169.00	73.00	36.00	15.00	55.00
HH?	700.00						
WTGB	140.00	163.00		2030.00	35.00	136.00	
WLB180		126.00	6069.00	1800.00			5.00

---- 594 PARAMETER TALL

	SAR	ELT	ATN	IOP	MILOP	MSAFE	MEP
HU25	3160	1187	279	4			528
HH65	2792	17069	447	74	418	47	510
HH3	3416	5131			1203	31	170
C130	1723	2048			229		
WPB82	12530	30988				675	357
WPB110	960	38500			4490		50
WTGB	2170	2233		2983	1120	136	
WLB180		1853	23855	1800	423	63	5
WLM157		1002	10997				
WLM133		1076	16375	481	770		548
WMEC270	681	33179			700		
WMEC210		42558			641		
TOTAL	27433	176826	51954	5342	9995	952	2168

APPENDIX D. FORTRAN LISTING

```
      PROGRAM USCG
*****************************************************************
*                                                               *
*   PROGRAM:  USER INTERFACE FOR COAST GUARD FORCE STRUCTURE    *
*             OPTIMIZATION MODEL                                *
*                                                               *
*   AUTHOR :  LT JOHN E TOMKO, USCG                             *
*                                                               *
*   WRITTEN:  13 MAY 1991                                       *
*                                                               *
*   LAST MODIFIED:  01 SEPTEMBER 1991                           *
*                                                               *
*   LANGUAGE:  FORTRAN                                          *
*                                                               *
*****************************************************************
* PROBLEM STATEMENT: A MENU-DRIVEN FRONT END USER INTERFACE FOR *
* MANIPULATING DATASETS CONTAINED IN SEPERATE FILES TO PRODUCE  *
* A GAMS FORMAT FOR DATA TO BE USED IN AN OPTIMIZATION MODEL.   *
*                                                               *
*****************************************************************
* EXTERNAL FILES:                                               *
*   INPUT:                                                      *
*          AMISS DATA A1 :  ATLANTIC AREA MISSION DATA          *
*          APLAT DATA A1 :  LANT AREA PLATFORM NAMES & DATA     *
*          ADIST DATA A1 :  ATLANTIC AREA DISTRICT DATA         *
*          AHOURS DATA A1:  LANT AREA RESOURCE HOUR REQUIREMENTS*
*          ATABLE DATA A1:  LANT AREA PARAMETER DATA            *
*          PMISS DATA A1 :  PACIFIC AREA MISSION DATA           *
*          PPLAT DATA A1 :  PAC AREA PLATFORM NAMES & DATA      *
*          PDIST DATA A1 :  PACIFIC AREA DISTRICT DATA          *
*          PHOURS DATA A1:  PAC AREA RESOURCE HOUR REQUIREMENTS *
*          PTABLE DATA A1:  PAC AREA PARAMETER DATA             *
*   OUTPUT:                                                     *
*          GAMS DATA A1  :  LISTING OF GAMS DATA FOR $INCLUDE   *
*****************************************************************
      INTEGER I, J, K, L, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      INTEGER INDP, LOCINDP(7)
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24), PCNT
      REAL MODEL(7)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER WORDA(4)*18, WORDB(4)*18, TYPE(24)*3, AREA*8
      LOGICAL GAMS
      GAMS = .FALSE.
      CALL INITIAL(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,CATA,
```

```
     +              CATB,VALA,VALB,ATOS,STOA,QTY)
      CALL GETDATA(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,MISSIONS,
     +              DISTRICT,PLATFORM,TYPE,VALA,CATA,CATB,VALB,ATOS,
     +              STOA,AREA,QTY,WORDA,WORDB,INDP,LOCINDP)
      CALL MAINMENU(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,MISSIONS,
     +              DISTRICT,PLATFORM,TYPE,CATA,CATB,VALA,VALB,ATOS,
     +              STOA,AREA,MODEL,QTY,WORDA,WORDB,GAMS,INDP,
     +              LOCINDP)
      CALL SAVEDATA(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,MISSIONS,
     +               DISTRICT,PLATFORM,TYPE,VALA,VALB,ATOS,STOA,AREA,
     +               QTY,WORDA,WORDB,CATA,CATB,INDP,LOCINDP)
      IF (GAMS) THEN
        WRITE(*,11)
      ELSE
        WRITE(*,22)
      ENDIF
11    FORMAT(/,' TO RUN MODEL TYPE: GAMS FLEET')
22    FORMAT(/,' NO GAMS MODEL WAS GENERATED DURING THIS SESSION')
      STOP
      END


      SUBROUTINE GETDATA(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,
     +                   MISSIONS,DISTRICT,PLATFORM,TYPE,VALA,CATA,
     +                   CATB,VALB,ATOS,STOA,AREA,QTY,WORDA,WORDB,
     +                   INDP,LOCINDP)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  GET DATA FROM EXTERNAL FILES                           *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  19 MAY 1991                                           *
*                                                                       *
*   MODIFIED   :  09 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE:  READ THE EXTERNAL FILES INTO ARRAYS FOR USE WITH THE       *
*  INTERFACE.                                                           *
*                                                                       *
*************************************************************************
      INTEGER I, J, K, L, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      INTEGER INDP, LOCINDP(7)
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER WORDA(4)*18, WORDB(4)*18, TYPE(24)*3, T*1, AREA*8
      LOGICAL WRONG
      WRONG = .TRUE.
      DO 5 WHILE (WRONG)
        WRITE(*,111)
        READ(*,99) T
```

```
        IF ((T .EQ. 'P') .OR. (T .EQ. 'A')) WRONG = .FALSE.
5     ENDDO
      OPEN(21,FILE='/'//T//'MISS DATA A1')
      OPEN(22,FILE='/'//T//'PLAT DATA A1')
      OPEN(23,FILE='/'//T//'DIST DATA A1')
      OPEN(24,FILE='/'//T//'HOURS DATA A1')
      OPEN(26,FILE='/'//T//'TABLE DATA A1')
      PRINT*,'RETRIEVING MISSION DATA...'
      READ(21,11) NUMISS
      DO 10 I = 1, NUMISS
        READ(21,22) MISSIONS(I)
10    CONTINUE
      PRINT*,'RETRIEVING PLATFORM DATA...'
      DO 15 J = 2,4
        NUMPLAT(J) = 0
15    CONTINUE
      READ(22,11) NUMPLAT(1)
      DO 20 I = 1, NUMPLAT(1)
        READ(22,44) PLATFORM(I),TYPE(I),HRS(I),COST(I)
        IF (TYPE(I) .EQ. 'IDS') THEN
          NUMPLAT(2) = NUMPLAT(2) + 1
        ELSEIF (TYPE(I) .EQ. 'IDA') THEN
          NUMPLAT(3) = NUMPLAT(3) + 1
        ELSE
          NUMPLAT(4) = NUMPLAT(4) + 1
        ENDIF
20    CONTINUE
      PRINT*,'RETRIEVING DISTRICT DATA...'
      READ(23,122) AREA
      READ(23,11) NUMDIST
      DO 30 I = 1, NUMDIST
        READ(23,33) DISTRICT(I)
30    CONTINUE
      READ(23,11) INDP
      DO 35 I = 1, INDP
       READ(23,11) LOCINDP(I)
35    CONTINUE
      PRINT*,'RETRIEVING PARAMETER DATA...'
      DO 40 I = 1, NUMDIST
        DO 50 J = 1, NUMISS
          READ(24,55) (HOURS(I,J,K),K=1,6)
50      CONTINUE
40    CONTINUE
      READ(26,77) (VALA(I), I=1,4)
      DO 80 I = 1,4
        READ(26,133) WORDA(I)
80    CONTINUE
      DO 90 I = 1, NUMPLAT(1)
        READ(26,88) (CATB(I,J), J=1,3)
90    CONTINUE
      READ(26,77) (VALB(I), I=1,4)
      DO 100 I = 1,4
```

```
        READ(26,133) WORDB(I)
100   CONTINUE
      DO 110 I = 1, NUMPLAT(1)
        READ(26,88) (CATA(I,J), J=1,NUMISS)
110   CONTINUE
      DO 130 I = 1, NUMPLAT(1)
        READ(26,88) (QTY(I,J), J=1,NUMDIST)
130   CONTINUE
      READ(26,77) (STOA(I), I=1,NUMISS)
      READ(26,77) (ATOS(I), I=1,NUMISS)
11    FORMAT(I3)
22    FORMAT(1X,A5)
33    FORMAT(1X,A10)
44    FORMAT(1X,A8,A3,1X,I4,1X,F7.4)
55    FORMAT(1X,6(I6,1X))
77    FORMAT(10(F4.2,1X))
88    FORMAT(7(I4,1X))
99    FORMAT(A1)
111   FORMAT(' ','PLEASE CHOOSE WHICH AREA DATA YOU WISH TO USE:',/,
     +       ' (A)TLANTIC',/,' (P)ACIFIC',/,' ENTER A OR P:')
122   FORMAT(1X,A8)
133   FORMAT(1X,A18)
      RETURN
      END


      SUBROUTINE SAVEDATA(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,
     +                    MISSIONS,DISTRICT,PLATFORM,TYPE,VALA,VALB,
     +                    ATOS,STOA,AREA,QTY,WORDA,WORDB,CATA,CATB,
     +                    INDP,LOCINDP)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  SAVE CHANGES TO DATABASES                              *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  19 MAY 1991                                           *
*                                                                       *
*   MODIFIED   :  09 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE:  IF SO DESIRED, STORE CHANGES TO THE DATABASE IN THE        *
*  ORIGINAL INPUT FILES FOR FUTURE USE.                                 *
*                                                                       *
*************************************************************************
      INTEGER I, J, K, L, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      INTEGER INDP, LOCINDP(7)
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER WORDA(4)*18, WORDB(4)*18, TYPE(24)*3, ANS*1, AREA*8
      REWIND(21)
```

```
      REWIND(22)
      REWIND(23)
      REWIND(24)
      REWIND(26)
      WRITE(*,11)
      READ(*,22) ANS
      IF (ANS .EQ. 'S') THEN
        PRINT*,'SAVING MISSION DATA...'
        WRITE(21,33) NUMISS
        DO 10 I = 1, NUMISS
          WRITE(21,77) MISSIONS(I)
10      CONTINUE
        PRINT*,'SAVING PLATFORM DATA...'
        WRITE(22,33) NUMPLAT(1)
        DO 20 I = 1, NUMPLAT(1)
          WRITE(22,99) PLATFORM(I),TYPE(I),HRS(I),COST(I)
20      CONTINUE
        PRINT*,'SAVING DISTRICT DATA...'
        WRITE(23,133) AREA
        WRITE(23,33) NUMDIST
        DO 30 I = 1, NUMDIST
          WRITE(23,88) DISTRICT(I)
30      CONTINUE
        WRITE(23,33) INDP
        DO 35 I = 1, INDP
          WRITE(23,33) LOCINDP(I)
35      CONTINUE
        PRINT*,'SAVING RESOURCE HOUR DATA...'
        DO 40 I = 1, NUMDIST
          DO 50 J = 1, NUMISS
            WRITE(24,55) (HOURS(I,J,K),K=1,6)
50        CONTINUE
40      CONTINUE
        PRINT*,'SAVING PARAMETER DATA...'
        WRITE(26,111) (VALA(I), I=1,4)
        DO 70 I = 1,4
          WRITE(26,144) WORDA(I)
70      CONTINUE
        DO 80 I = 1, NUMPLAT(1)
          WRITE(26,122) (CATB(I,J), J=1,3)
80      CONTINUE
        WRITE(26,111) (VALB(I), I=1,4)
        DO 90 I = 1,4
          WRITE(26,144) WORDB(I)
90      CONTINUE
        DO 100 I = 1, NUMPLAT(1)
          WRITE(26,122) (CATA(I,J), J=1,NUMISS)
100     CONTINUE
        DO 110 I = 1, NUMPLAT(1)
          WRITE(26,122) (QTY(I,J), J=1,NUMDIST)
110     CONTINUE
        WRITE(26,111) (STOA(I), I=1,NUMISS)
```

```
        WRITE(26,111) (ATOS(I), I=1,NUMISS)
      ELSE
        WRITE(*,44)
      ENDIF
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(26)
11    FORMAT(' ','EXITING PROGRAM.',/,' YOU MAY EITHER',/,' (S)AVE ',
     +       'CHANGES MADE DURING THIS SESSION',5X,'OR',/,' (E)XIT ',
     +       'WITHOUT CHANGING THE MASTER DATABASE.',/,
     +       ' PLEASE ENTER S OR E:')
22    FORMAT(A1)
33    FORMAT(I3)
44    FORMAT(' ','MASTER DATABASE NOT MODIFIED BY CHANGES MADE DURING ',
     +       'THIS SESSION.')
55    FORMAT(1X,6(I6,1X))
77    FORMAT(1X,A5)
88    FORMAT(1X,A10)
99    FORMAT(1X,A8,A3,1X,I4,1X,F7.4)
111   FORMAT(10(F4.2,1X))
122   FORMAT(7(I4,1X))
133   FORMAT(1X,A8)
144   FORMAT(' ',A18)
      RETURN
      END


      SUBROUTINE MAINMENU(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,
     +                    MISSIONS,DISTRICT,PLATFORM,TYPE,CATA,
     +                    CATB,VALA,VALB,ATOS,STOA,AREA,MODEL,QTY,
     +                    WORDA,WORDB,GAMS,PCNT,INDP,LOCINDP)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  MAIN MENU CONTROLLER                                   *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  20 MAY 1991                                           *
*                                                                       *
*   MODIFIED   :  04 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE:  PROVIDE USER WITH MAIN MENU WHICH CAN ACCESS ALL OTHER     *
*  MENUS AVAILABLE IN THE INTERFACE.                                    *
*                                                                       *
*************************************************************************
      INTEGER I, J, K, L, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      INTEGER INDP, LOCINDP(7), PICK, CNT
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24)
```

```
      REAL MODEL(7)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER WORDA(4)*18, WORDB(4)*18, TYPE(24)*3, ANS*1, OPT*1
      CHARACTER AREA*8
      LOGICAL GAMS
      ANS = 'A'
      PICK = 0
      CNT = 0
      DO 10 WHILE (ANS .NE. 'E')
        WRITE(*,11)
        READ(*,22) ANS
        IF (ANS .EQ. 'D') THEN
          PICK = 1
          CALL DEFINE(MODEL)
        ELSEIF (ANS .EQ. 'P') THEN
          CALL OUT(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,CNT,MISSIONS,
     +             DISTRICT,PLATFORM,TYPE,CATA,CATB,VALA,VALB,ATOS,
     +             STOA,QTY,PCNT,INDP,LOCINDP,WORDA,WORDB)
        ELSEIF (ANS .EQ. 'M') THEN
          OPT = 'A'
          DO 20 WHILE (OPT .NE. 'E')
            WRITE(*,33)
            READ(*,22) OPT
            IF (OPT .EQ. 'P') THEN
              CALL EDITPLAT(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,CATA,
     +                      CATB,MISSIONS,DISTRICT,PLATFORM,TYPE,QTY,
     +                      WORDA,WORDB,VALA,VALB)
            ELSEIF (OPT .EQ. 'D') THEN
              CALL EDITDIST(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,QTY,
     +                      MISSIONS,DISTRICT,PLATFORM,TYPE,AREA,INDP,
     +                      LOCINDP)
            ELSEIF (OPT .EQ. 'M') THEN
              CALL EDITMISS(NUMDIST,NUMPLAT,NUMISS,HOURS,MISSIONS,
     +                     DISTRICT,PLATFORM,ATOS,STOA,CATA,VALA,WORDA)
            ELSEIF (OPT .NE. 'E') THEN
              WRITE(*,55)
            ENDIF
20        ENDDO
        ELSEIF (ANS .EQ. 'G') THEN
          IF (PICK .NE. 0) THEN
            GAMS = .TRUE.
            CALL DOIT(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,MISSIONS,
     +                DISTRICT,PLATFORM,TYPE,CATA,CATB,VALA,VALB,
     +                ATOS,STOA,QTY,PCNT,INDP,LOCINDP,MODEL)
          ELSE
            WRITE(*,44)
          ENDIF
        ELSEIF (ANS .EQ. 'C') THEN
          CALL CHANGE(WORDA,WORDB,VALA,VALB,ATOS,STOA,NUMPLAT,
     +                NUMISS,MISSIONS,PLATFORM,PCNT)
        ELSEIF (ANS .NE. 'E') THEN
          WRITE(*,55)
```

```
        ENDIF
10     ENDDO
11     FORMAT(' ',20X,'MAIN MENU',/,' THE FOLLOWING OPTIONS ARE ',
     +       'AVAILABLE:',/,' (D)EFINE PROBLEM',/,' (P)RINT DATABASE'
     +       ,/,' (M)ODIFY DATABASE',/,' (G)ENERATE GAMS FILE',/,
     +       ' (C)HANGE PARAMETERS',/,' (E)XIT PROGRAM',/,
     +       ' ENTER YOUR CHOICE (D,P,M,G,C,E):')
22     FORMAT(A1)
33     FORMAT(' ',5X,'MODIFY DATABASE MENU',/,' SELECT DATABASE TO BE ',
     +       'MODIFIED:',/,' (P)LATFORM',/,' (D)ISTRICT',/,' (M)ISSION',
     +       /,' (E)XIT MODIFY DATABASE ROUTINE',/,
     +       ' ENTER YOUR CHOICE (P,D,M,E):')
44     FORMAT(' ','*** ERROR ***',/,' (D)EFINE PROBLEM FIRST',/)
55     FORMAT(' ','*** ERROR ***',/,' INVALID OPTION',/)
       RETURN
       END


       SUBROUTINE INITIAL(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,CATA,
     +                    CATB,VALA,VALB,ATOS,STOA,QTY)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  INITIALIZE ARRAYS                                      *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  25 MAY 1991                                           *
*                                                                       *
*   MODIFIED   :  15 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE:  INITIALIZE VARIABLES TO PROTECT AGAINST BAD DATA BEING     *
*  MISTAKENLY ENTERED IN THE ARRAYS.                                    *
*                                                                       *
*************************************************************************
       INTEGER I, J, K, L, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
       INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
       REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24)
       NUMISS = 0
       DO 10 I = 1,24
         HRS(I) = 0
         COST(I) = 0.0
         DO 20 J = 1,10
           CATA(I,J) = 1
20       CONTINUE
         DO 30 J = 1,3
           CATB(I,J) = 1
30       CONTINUE
         DO 40 J = 1,7
           QTY(I,J) = 0
40       CONTINUE
10     CONTINUE
```

```
      DO 50 I = 1,4
        NUMPLAT(I) = 0
        VALA(I) = 0.0
        VALB(I) = 0.0
50    CONTINUE
      DO 60 I = 1,10
        ATOS(I) = 0.0
        STOA(I) = 0.0
60    CONTINUE
      NUMDIST = 0
      DO 80 I = 1,7
        DO 90 J = 1,10
          DO 100 K = 1,6
            HOURS(I,J,K) = 0
100       CONTINUE
90      CONTINUE
80    CONTINUE
      RETURN
      END


      SUBROUTINE DEFINE(MODEL)
*****************************************************************
*                                                               *
*   SUBROUTINE:  DEFINE FLEET MIX PROBLEM                       *
*                                                               *
*   AUTHOR     :  LT J.E. TOMKO, USCG                           *
*                                                               *
*   WRITTEN    :  04 JUNE 1991                                  *
*                                                               *
*   MODIFIED   :  01 SEPTEMBER 1991                             *
*                                                               *
*****************************************************************
*  PURPOSE: ALLOW THE SUER TO SPECIFY CERTAIN ASPECTS OF THE PROBLEM *
*  SUCH AS COSTS TO BE CONSIDERED.                              *
*                                                               *
*****************************************************************
      CHARACTER ANS*1, OPT*1
      REAL MODEL(7), TEMP
      LOGICAL ERROR
      ERROR = .TRUE.
      DO 10 WHILE (ERROR)
        WRITE(*,11)
        READ(*,22) OPT
        ERROR = .FALSE.
        IF (OPT .EQ. 'A') THEN
          MODEL(1) = 1.0
        ELSEIF (OPT .EQ. 'B') THEN
          MODEL(1) = 3.0
          MODEL(2) = 0.0
          MODEL(4) = 0.0
          MODEL(6) = 0.0
```

```
        ELSEIF (OPT .EQ. 'C') THEN
          MODEL(1) = 2.0
          MODEL(3) = 0.0
          MODEL(5) = 0.0
          MODEL(7) = 0.0
        ELSE
         WRITE(*,33)
         ERROR = .TRUE.
        ENDIF
10    ENDDO
      IF ((OPT .EQ. 'A') .OR. (OPT .EQ. 'C')) THEN
        WRITE(*,55)
        READ(*,*) MODEL(6)
        MODEL(6) = MODEL(6) / 1000.0
        IF (MODEL(6) .LT. 0.0) MODEL(6) = 0.0
        WRITE(*,44)
        READ(*,*) TEMP
        IF (TEMP .GE. 0.0) THEN
          MODEL(2) = 1.0
          MODEL(4) = TEMP / 1000.0
        ELSE
          MODEL(2) = 0.0
          MODEL(4) = 0.0
        ENDIF
      ENDIF
      IF ((OPT .EQ. 'A') .OR. (OPT .EQ. 'B')) THEN
        WRITE(*,77)
        READ(*,*) MODEL(7)
        MODEL(7) = MODEL(7) / 1000.0
        IF (MODEL(7) .LT. 0.0) MODEL(7) = 0.0
        WRITE(*,66)
        READ(*,*) TEMP
        IF (TEMP .GE. 0.0) THEN
          MODEL(3) = 1.0
          MODEL(5) = TEMP / 1000.0
        ELSE
          MODEL(3) = 0.0
          MODEL(5) = 0.0
        ENDIF
      ENDIF
      WRITE(*,88)
      IF (OPT .EQ. 'A') THEN
        WRITE(*,99)
      ELSEIF(OPT .EQ. 'B') THEN
        WRITE(*,122)
      ELSE
        WRITE(*,111)
      ENDIF
      IF (MODEL(2) .EQ. 0.0) THEN
        WRITE(*,133)
        WRITE(*,177) INT(MODEL(6)*1000000.0)
      ELSE
```

```
        WRITE(*,144) INT(MODEL(6)*1000000.0),INT(MODEL(4)*1000000.0)
      ENDIF
      IF (MODEL(3) .EQ. 0.0) THEN
        WRITE(*,155)
        WRITE(*,188) INT(MODEL(7)*1000000.0)
      ELSE
        WRITE(*,166) INT(MODEL(7)*1000000.0),INT(MODEL(5)*1000000.0)
      ENDIF
11    FORMAT(' ','THE FOLLOWING MODELS ARE AVAILABLE:',/,
     +       ' (A) FULL MODEL WITH ACQUISITION AND MOVEMENT COSTS',/,
     +       ' (B) MODEL WITHOUT ACQUISITION COST',/,' (C) MODEL ',
     +       'WITHOUT MOVEMENT COST',/,' ENTER CHOICE (A,B,C):')
22    FORMAT(A1)
33    FORMAT(' ','*** ERROR ***',/,' INVALID OPTION',/)
44    FORMAT(' ','ENTER UPPER LIMIT FOR TOTAL ACQUISITION COST:',/,
     +           ' ENTER A (-1) FOR NO LIMIT.  ENTER VALUE IN ',
     +           'THOUSANDS OF DOLLARS:')
55    FORMAT(' ','ENTER ACQUISITION PENALTY TO USE IN THIS MODEL',/,
     +           ' (IN THOUSANDS OF DOLLARS (1 MILLION = 1000.0):')
66    FORMAT(' ','ENTER UPPER LIMIT FOR TOTAL MOVEMENT COST:',/,
     +           ' ENTER A (-1) FOR NO LIMIT.  ENTER VALUE IN ',
     +           'THOUSANDS OF DOLLARS:')
77    FORMAT(' ','ENTER MOVEMENT PENALTY TO USE IN THIS MODEL',/,
     +           ' (IN THOUSANDS OF DOLLARS (1 MILLION = 1000.0):')
88    FORMAT(' ','THE FOLLOWING MODEL HAS BEEN DEFINED:',/)
99    FORMAT(' ','FULL MODEL WITH ACQUISITION AND MOVEMENT COSTS.')
111   FORMAT(' ','MODEL WITH ACQUISITION COSTS; NO MOVEMENT COSTS.')
122   FORMAT(' ','MODEL WITH MOVEMENT COSTS; NO ACQUISITION COSTS.')
133   FORMAT(' ','THERE IS NO TOTAL ACQUISITION COST LIMIT.')
144   FORMAT(' ','INDIVIDUAL ACQUISITION COST IS $',I9,/,' A TOTAL',
     +           ' ACQUISITION COST LIMIT OF $',I9,' IS ENFORCED.')
155   FORMAT(' ','THERE IS NO TOTAL MOVEMENT COST LIMIT.',/)
166   FORMAT(' ','INDIVIDUAL MOVEMENT COST IS $',I9,/,' A TOTAL',
     +           ' MOVEMENT COST LIMIT OF $',I9,' IS ENFORCED.',/)
177   FORMAT(' ','INDIVIDUAL ACQUISITION COST IS $',I9)
188   FORMAT(' ','INDIVIDUAL MOVEMENT COST IS $',I9)
      RETURN
      END
         SUBROUTINE OUT(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,CNT,
        +               MISSIONS,DISTRICT,PLATFORM,TYPE,CATA,CATB,VALA,
        +               VALB,ATOS,STOA,QTY,PCNT,INDP,LOCINDP,WORDA,WORDB)
   ************************************************************************
   *                                                                      *
   *   SUBROUTINE:  OUTPUT DATABASE TO FILE OR SCREEN                     *
   *                                                                      *
   *   AUTHOR     :  LT J.E. TOMKO, USCG                                  *
   *                                                                      *
   *   WRITTEN    :  12 JUNE 1991                                         *
   *                                                                      *
   *   MODIFIED   :  12 JUNE 1991                                         *
   *                                                                      *
   ************************************************************************
```

```
*  PURPOSE:  OUTPUT THE DATABASE IN A USER FRIENDLY FORMAT TO A FILE *
*  OR TO THE SCREEN.                                                 *
**********************************************************************
      INTEGER  NUMDIST, NUMPLAT(4), NUMISS, HRS(24), INDP, LOCINDP(7)
      INTEGER HOURS(7,10,6), QTY(24,7), FILE
      INTEGER CNT, CATB(24,3), CATA(24,10)
      REAL COST(24), PCNT
      REAL VALA(4), VALB(4), ATOS(10), STOA(10)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER TYPE(24)*3, ANS*1, WORDA(4)*18, WORDB(4)*18
      CHARACTER CODE(3)*3, STR(3)*16
      LOGICAL ERROR
      DATA STR/'DISTRICT SURFACE','DISTRICT AIR    ','AREA SURFACE    '/
      DATA CODE/'IDS', 'IDA', 'IAS'/
      ERROR = .TRUE.
      IF (CNT .GE. 1) THEN
        WRITE(*,55)
      ENDIF
      DO 10 WHILE (ERROR)
        PRINT*,'OUTPUT TO (F)ILE OR (S)CREEN:'
        READ(*,11) ANS
        IF (ANS .EQ. 'F') THEN
          IF (CNT .GE. 1) THEN
            WRITE(*,66)
          ENDIF
          ERROR = .FALSE.
          FILE = 31
          OPEN(31,FILE='/DATA LISTING A1')
          CNT = CNT + 1
        ELSEIF (ANS .EQ. 'S') THEN
          ERROR = .FALSE.
          FILE = 6
        ELSE
          WRITE(*,22) ANS
        ENDIF
10    ENDDO
      ERROR = .TRUE.
      DO 20 WHILE (ERROR)
        WRITE(*,33)
        READ(*,11) ANS
        IF (ANS .EQ. 'M') THEN
          CALL OUTMISS(FILE,HOURS,MISSIONS,DISTRICT,NUMDIST,NUMISS)
        ELSEIF (ANS .EQ. 'P') THEN
          CALL OUTPLAT(FILE,NUMPLAT,PLATFORM,TYPE,COST,QTY,CATA,
     +                 CATB,VALA,VALB,WORDA,WORDB,STR,CODE,NUMISS,
     +                 MISSIONS,NUMDIST,DISTRICT,HRS)
        ELSEIF (ANS .EQ. 'D') THEN
          CALL OUTDIST(FILE,HOURS,MISSIONS,DISTRICT,NUMDIST,
     +                 NUMISS,PLATFORM,QTY,TYPE,STR,CODE,INDP,
     +                 LOCINDP,NUMPLAT)
        ELSEIF (ANS .EQ. 'T') THEN
          CALL OUTPAR(FILE,ATOS,STOA,NUMISS,MISSIONS,VALA,VALB,WORDA,
```

```
     +                WORDB,PCNT)
         ELSEIF (ANS .EQ. 'A') THEN
           CALL OUTMISS(FILE,HOURS,MISSIONS,DISTRICT,NUMDIST,NUMISS)
           CALL OUTPLAT(FILE,NUMPLAT,PLATFORM,TYPE,COST,QTY,CATA,
     +                CATB,VALA,VALB,WORDA,WORDB,STR,CODE,NUMISS,
     +                MISSIONS,NUMDIST,DISTRICT,HRS)
           CALL OUTDIST(FILE,HOURS,MISSIONS,DISTRICT,NUMDIST,
     +                NUMISS,PLATFORM,QTY,TYPE,STR,CODE,INDP,
     +                LOCINDP,NUMPLAT)
           CALL OUTPAR(FILE,ATOS,STOA,NUMISS,MISSIONS,VALA,VALB,WORDA,
     +                WORDB,PCNT)
         ELSEIF (ANS .NE. 'E') THEN
           WRITE(*,22) ANS
         ELSE
           ERROR = .FALSE.
         ENDIF
20     ENDDO
       IF (FILE .EQ. 31) THEN
         WRITE(*,44)
         CLOSE(31)
       ENDIF
11     FORMAT(A1)
22     FORMAT(' ','*** ERROR ***',/,' (',A1,') IS NOT A VALID ENTRY',/)
33     FORMAT(' ',17X,'OUTPUT MENU',/,
     +       ' THE FOLLOWING DATA ARE AVAILABLE FOR OUTPUT:',/,
     +       ' (D)ISTRICT DATA',/,' (P)LATFORM DATA',/,' (M)ISSION ',
     +       'DATA',/,' (T)ABLE DATA',/,' (A)LL DATA',/,' (E)XIT OUT',
     +       'PUT ROUTINE',/,' ENTER YOUR CHOICE (D,P,M,T,A, OR E):')
44     FORMAT(' ','OUTPUT SENT TO FILE: DATA LISTING A1',/)
55     FORMAT(' ','*** WARNING ***',/,' YOU HAVE ALREADY SENT OUTPUT ',
     +       'TO THE FILE.',/,' SELECTING (F)ILE OUTPUT WILL ERASE ',
     +       'THE LAST FILES PRINTED.',/)
66     FORMAT(' ','(F)ILE OUTPUT SELECTED',/,' OVERWRITING PREVIOUS ',
     +       'OUTPUT.',/)
       RETURN
       END


       SUBROUTINE OUTMISS(FILE,HOURS,MISSIONS,DISTRICT,NUMDIST,NUMISS)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  WRITE MISSION DATA TO SCREEN OR FILE                   *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  12 JUNE 1991                                          *
*                                                                       *
*   MODIFIED   :  12 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE: WRITE THE MISSION DATA TO OUTFILE (SCREEN OR FILE) IN A     *
*  READABLE FORM FOR USER.                                              *
```

```
*********************************************************************
      INTEGER I, J, K, FILE, NUMDIST, NUMISS, HOURS(7,10,6)
      INTEGER TOTHRS, GRAND(7), CNT
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10
      CNT = 100
      DO 10 I = 1, NUMISS
        DO 20 K = 1,7
          GRAND(K) = 0
20      CONTINUE
        IF (CNT .GE. 63-NUMDIST) THEN
          WRITE(FILE,11) MISSIONS(I)
          CNT = 5
        ELSE
          WRITE(FILE,44) MISSIONS(I)
          CNT = CNT + 5
        ENDIF
        DO 30 J = 1, NUMDIST
          TOTHRS = 0
          DO 40 K = 1,6
           TOTHRS = TOTHRS + HOURS(J,I,K)
40        CONTINUE
          WRITE(FILE,22) DISTRICT(J), (HOURS(J,I,K), K=1,6), TOTHRS
          CNT = CNT + 1
          GRAND(7) = GRAND(7) + TOTHRS
          DO 50 K = 1,6
            GRAND(K) = GRAND(K) + HOURS(J,I,K)
50        CONTINUE
30      CONTINUE
        WRITE(FILE,33) (GRAND(K), K=1,7)
        CNT = CNT + 2
10    CONTINUE
11    FORMAT('1','MISSION: ',A5,/,12X,'SURFACE HOURS REQUIRED',6X,
     +       'AIR HOURS REQUIRED',/,15X,'CLASS',21X,'CLASS',/,11X,
     +       'DISTRICT',7X,'1',6x,'2',6x,'3',10x,
     +       '1',7X,'2',5X,'3',8X,'TOTAL',/,1X,
     +       8('-'),2X,3(2X,5('-')),4X,3(2X,5('-')),6X,6('-'))
22    FORMAT(' ',A10,2X,3(I5,2X),4X,3(I5,2X),3X,I7)
33    FORMAT(' ',12X,5('-'),2(2X,5('-')),6X,3(5('-'),2X),4X,
     +       6('-'),/,1X,'TOTAL',6X,3(I6,1X),4X,3(I6,1X),3X,I8)
44    FORMAT(/,' MISSION: ',A5,/,12X,'SURFACE HOURS REQUIRED',6X,
     +       'AIR HOURS REQUIRED',/,15X,'CLASS',
     +       10X,'CLASS',/,1X,'DISTRICT',7X,
     +       '1',6X,'2',6X,'3',10X,'1',7X,'2',5X,'3',8X,'TOTAL',/,
     +       1X,8('-'),2X,3(2X,5('-')),4X,3(2X,5('-')),6X,6('-'))
      RETURN
      END


      SUBROUTINE OUTPLAT(FILE,NUMPLAT,PLATFORM,TYPE,COST,QTY,CATA,
     +                   CATB,VALA,VALB,WORDA,WORDB,STR,CODE,NUMISS,
     +                   MISSIONS,NUMDIST,DISTRICT,HRS)
*********************************************************************
```

```
*                                                                      *
*   SUBROUTINE:  OUTPUT PLATFORM DATA                                  *
*                                                                      *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                  *
*                                                                      *
*   WRITTEN    :  12 JUNE 1991                                         *
*                                                                      *
*   MODIFIED   :  12 JUNE 1991                                         *
*                                                                      *
************************************************************************
*  PURPOSE: WRITE THE PLATFORM DATA TO OUTFILE (SCREEN OR FILE) IN A    *
*  READABLE FORMAT.                                                    *
************************************************************************
      INTEGER I, J, K, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER QTY(24,7), CATB(24,3), CATA(24,10), FILE, CNT
      REAL VALA(4), VALB(4), COST(24)
      CHARACTER MISSIONS(10)*5, PLATFORM(24)*8, STR(3)*16
      CHARACTER CODE(3)*3, TYPE(24)*3, WORDA(4)*18, WORDB(4)*18
      CHARACTER DISTRICT(7)*10
      CNT = 100
      DO 10 I = 1, NUMPLAT(1)
        DO 20 J = 1,3
          IF (TYPE(I) .EQ. CODE(J)) K = J
20      CONTINUE
        IF (CNT .GE. 55-(NUMMISS+INT(NUMDIST/4)+1)) THEN
          WRITE(FILE,11) PLATFORM(I),STR(K),COST(I),HRS(I)
          CNT = 5
        ELSE
          WRITE(FILE,22) PLATFORM(I),STR(K),COST(I),HRS(I)
          CNT = CNT + 5
        ENDIF
        WRITE(FILE,33)
        CNT = CNT + 4
        DO 30 J = 1, NUMISS
          WRITE(FILE,44) MISSIONS(J),WORDA(CATA(I,J)),
     +                   INT((VALA(CATA(I,J))*100.0)+0.5)
          CNT = CNT + 1
30      CONTINUE
        WRITE(FILE,55)
        CNT = CNT + 4
        DO 40 J = 1, 3
          WRITE(FILE,66) J,WORDB(CATB(I,J)),
     +                   INT((VALB(CATB(I,J))*100.0)+0.5)
          CNT = CNT + 1
40      CONTINUE
        WRITE(FILE,77) PLATFORM(I)
        CNT = CNT + 2
        DO 50 K = 1, (NUMDIST/3) + 1
          WRITE(FILE,88) (DISTRICT(J),QTY(I,J), J=(K-1)*3+1,
     +                                MIN((K-1)*3+3,NUMDIST))
          CNT = CNT + 1
50      CONTINUE
```

```
10     CONTINUE
11     FORMAT('1',/,' PLATFORM NAME: ',A8,3X,'TYPE: ',A16,/,10X,'YEARLY'
     +       ,' OPERATING DATA:',/,6X,'COST: ',F6.2,7X,'HOURS: ',I4,/)
22     FORMAT(//,' PLATFORM NAME: ',A8,3X,'TYPE: ',A16,/,10X,'YEARLY ',
     +       ' OPERATING DATA:',/,6X,'COST: ',F6.2,7X,'HOURS: ',I4,/)
33     FORMAT(' ','MISSION PROFILE DATA',/,1X,20('='),/,' MISSION',
     +       4X,'DESCRIPTION',9X,'PERCENTAGE',/,1X,7('-'),2X,16('-'),
     +       6X,10('-'))
44     FORMAT(' ',A5,4X,A18,6X,I3,'%')
55     FORMAT(/,' CLASS PROFILE DATA',,/,1X,18('='),/,' CLASS',7x,
     +       'DESCRIPTION',9X,'PERCENTAGE',/,1X,9('-'),2X,16('-'),
     +       6X,10('-'))
66     FORMAT(' ',4X,I1,6X,A18,6X,I3,'%')
77     FORMAT(/,' CURRENT LOCATION OF ALL ',A8,' PLATFORMS:')
88     FORMAT(' ',3X,3(A10,2X,I3,5X))
       RETURN
       END


       SUBROUTINE OUTDIST(FILE,HOURS,MISSIONS,DISTRICT,NUMDIST,
     +                    NUMISS,PLATFORM,QTY,TYPE,STR,CODE,INDP,
     +                    LOCINDP,NUMPLAT)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  WRITE DISTRICT DATA TO SCREEN OR FILE                  *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  12 JUNE 1991                                          *
*                                                                       *
*   MODIFIED   :  12 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE: WRITE DISTRICT DATA TO OUTFILE (SCREEN OR FILE) IN A        *
*  READABLE FORMAT.                                                     *
*************************************************************************
       INTEGER  I, J, K, NUMDIST, NUMPLAT(4), NUMISS
       INTEGER HOURS(7,10,6), QTY(24,7), FILE, INDP, LOCINDP(7)
       INTEGER TOTHRS, GRAND(7), CNT
       CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
       CHARACTER CODE(3)*3, STR(3)*16, TYPE(24)*3
       CNT = 100
       DO 10 I = 1, NUMDIST
         DO 20 K = 1,7
           GRAND(K) = 0
20       CONTINUE
         IF (CNT .GE. 59-(NUMMISS+INT(NUMPLAT(1)/4)+1)) THEN
           WRITE(FILE,11) DISTRICT(I)
           CNT = 5
         ELSE
           WRITE(FILE,44) DISTRICT(I)
           CNT = CNT + 5
```

```
        ENDIF
        DO 30 J = 1, NUMISS
          TOTHRS = 0
          DO 40 K = 1,6
           TOTHRS = TOTHRS + HOURS(I,J,K)
40        CONTINUE
          WRITE(FILE,22) MISSIONS(J), (HOURS(I,J,K), K=1,6), TOTHRS
          CNT = CNT + 1
          GRAND(7) = GRAND(7) + TOTHRS
          DO 50 K = 1,6
            GRAND(K) = GRAND(K) + HOURS(I,J,K)
50        CONTINUE
30      CONTINUE
        WRITE(FILE,33) (GRAND(K), K=1,7)
        CNT = CNT + 2
        WRITE(FILE,55) DISTRICT(I)
        CNT = CNT + 4
        DO 60 K = 1, (NUMPLAT(1)/4) + 1
          WRITE(FILE,66) (PLATFORM(J),QTY(J,I), J=(K-1)*4+1,
     +                                  MIN((K-1)*4+4,NUMPLAT(1)))
        CNT = CNT + 1
60      CONTINUE
10     CONTINUE
11     FORMAT('1','DISTRICT: ',A10,/,12X,'SURFACE HOURS REQUIRED',6X,
     +       'AIR HOURS REQUIRED',/,15X,'CLASS',
     +       10X,'CLASS',/,1X,'MISSION',8X,
     +       '1',6X,'2',6X,'3',10X,'1',7X,'2',5X,'3',8X,'TOTAL',/,
     +       1X,8('-'),2X,3(2X,5('-')),4X,3(2X,5('-')),6X,6('-'))
22     FORMAT(' ',A5,7X,3(I5,2X),4X,3(I5,2X),3X,I7)
33     FORMAT(' ',12X,5('-'),2(2X,5('-')),6X,3(5('-'),2X),4X,
     +       6('-'),/,1X,'TOTAL',6X,3(I6,1X),4X,3(I6,1X),3X,I8)
44     FORMAT(/,' DISTRICT: ',A10,/,12X,'SURFACE HOURS REQUIRED',6X,
     +       'AIR HOURS REQUIRED',/,15X,'CLASS',
     +       10X,'CLASS',/,1X,'MISSION',8X,
     +       '1',6X,'2',6X,'3',10X,'1',7X,'2',5X,'3',8X,'TOTAL',/,
     +       1X,8('-'),2X,3(2X,5('-')),4X,3(2X,5('-')),6X,6('-'))
55     FORMAT(/,' CURRENT PLATFORMS LOCATED WITHIN ',A10,' DISTRICT ',
     +       'BOUNDARIES:',/,1X,4('NAME',8X,'QTY',3X),/,1X,
     +       4(9('-'),3X,3('-'),3X))
66     FORMAT(' ',4(A8,5X,I2,3X))
       RETURN
       END
```

```
      SUBROUTINE OUTPAR(FILE,ATOS,STOA,NUMISS,MISSIONS,VALA,VALB,
     +                  WORDA,WORDB,PCNT)
*********************************************************************
*                                                                   *
*   SUBROUTINE:  WRITE PARAMETER DATA TO SCREEN OR FILE             *
*                                                                   *
*   AUTHOR     :  LT J.E. TOMKO, USCG                               *
*                                                                   *
*   WRITTEN    :  12 JUNE 1991                                      *
*                                                                   *
*   MODIFIED   :  12 JUNE 1991                                      *
*                                                                   *
*********************************************************************
*  PURPOSE: WRITE THE PARAMETER DATA TO OUTFILE (SCREEN OR FILE) IN  *
*  A READABLE FORMAT.                                               *
*********************************************************************
      INTEGER I, J, K, FILE, NUMISS
      REAL VALA(4), VALB(4), ATOS(10), STOA(10)
      CHARACTER MISSIONS(10)*5, WORDA(4)*18, WORDB(4)*18
      WRITE(FILE,11) (MISSIONS(I), I=1,NUMISS)
      WRITE(FILE,33) (INT(ATOS(I)*100.0+0.5), I=1,NUMISS)
      WRITE(FILE,22) (MISSIONS(I), I=1,NUMISS)
      WRITE(FILE,33) (INT(STOA(I)*100.0+0.5), I=1,NUMISS)
      WRITE(FILE,44)
      WRITE(FILE,66)
      DO 10 I = 1,4
       WRITE(FILE,77) I, INT(VALA(I)*100.0), WORDA(I)
10    CONTINUE
      WRITE(FILE,55)
      WRITE(FILE,66)
      DO 20 I = 1,4
       WRITE(FILE,77) I, INT(VALB(I)*100.0), WORDB(I)
20    CONTINUE
11    FORMAT('1','AIRCRAFT CAN CURRENTLY DO THE FOLLOWING AMOUNT',/,
     +       ' OF SURFACE RESOURCE HOURS IN EACH MISSION AREA:',/,
     +       4X,10(A5,2X))
22    FORMAT(//,' CUTTERS CAN CURRENTLY DO THE FOLLOWING AMOUNT',/,
     +       ' OF AIR RESOURCE HOURS IN EACH MISSION AREA:',/,
     +       4X,10(A5,2X))
33    FORMAT(' ',2X,10(I3,'%',3X),/)
44    FORMAT(//,' THE FOLLOWING VALUES AND DESCRIPTIONS ARE DEFINED ',
     +       'FOR',/,' MISSION ASSIGNMENT CODES:')
55    FORMAT(//,' THE FOLLOWING VALUES AND DESCRIPTIONS ARE DEFINED ',
     +       'FOR',/,' CLASS CODES:')
66    FORMAT(' ',5X,'CODE',3X,'VALUE',8X,'DESCRIPTION',/,6X,4('-'),
     +       3X,5('-'),4X,18('-'))
77    FORMAT(' ',7X,I1,5X,I3,'%',5X,A18)
      RETURN
      END
```

```
      SUBROUTINE EDITPLAT(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,CATA,
     +                    CATB,MISSIONS,DISTRICT,PLATFORM,TYPE,QTY,
     +                    WORDA,WORDB,VALA,VALB)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  EDIT PLATFORM LISTING                                *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  16 MAY 1991                                         *
*                                                                       *
*     MODIFIED   :  30 AUGUST 1991                                      *
*************************************************************************
* PURPOSE:  MENU DRIVEN USER INTERFACE TO ALLOW FOR MODIFICATION OF     *
* THE COAST GUARD PLATFORM DATA. USER MAY ADD, DELETE, CHANGE, AND      *
* LIST THE CURRENT PLATFORMS ON FILE. THE MAXIMUM NUMBER OF PLATFORMS   *
* WAS SET AT 24 TOTAL. THIS IS TO AVOID EXTREMELY LARGE GAMS MODELS.    *
* ROUTINE IS WRITTEN WITH ERROR CHECKING CODE TO AVOID DUPLICATION      *
* OF PLATFORM NAMES OR ATTEMPTING DELETION OF NONEXISTING NAMES.        *
*************************************************************************
      INTEGER I, J, K, NUMPLAT(4), LOCATE, FINDP, HOLD, NUMDIST
      INTEGER NUMISS, HRS(24), HOURS(7,10,6), QTY(24,7)
      INTEGER CATB(24,3), CATA(24,10), PICK, COUNT, LOOP
      REAL COST(24), VALA(4), VALB(4)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, CODE(4)*3
      CHARACTER PLATFORM(24)*8, TEMP*8, ANS*1, TYPE(24)*3, CG*26
      CHARACTER WORDA(4)*18, WORDB(4)*18
      DATA CODE/'   ','IDS','IDA','IAS'/
      PICK = 0
      DO 30 WHILE (PICK .EQ. 0)
        WRITE(*,155)
        READ(*,66) ANS
        IF (ANS .EQ. 'A') THEN
          CG = 'DISTRICT SURFACE PLATFORMS'
          PICK = 2
        ELSEIF (ANS .EQ. 'B') THEN
          CG = 'DISTRICT AIR PLATFORMS    '
          PICK = 3
        ELSEIF (ANS .EQ. 'C') THEN
          CG = 'AREA SURFACE PLATFORMS    '
          PICK = 4
        ELSEIF (ANS .EQ. 'D') THEN
          CG = 'COAST GUARD PLATFORMS     '
          PICK = 1
        ELSE
          WRITE(*,166) ANS
        ENDIF
30    ENDDO
      WRITE (*,22) NUMPLAT(PICK), CG
      IF (PICK .EQ. 1) THEN
        LOOP = NUMPLAT(1)/6
        DO 10 I = 1, LOOP
```

```
          WRITE(*,199) (PLATFORM(J), J=1+(I-1)*6,6+(I-1)*6)
10      CONTINUE
        IF (MOD(NUMPLAT(1),6) .NE. 0) THEN
          WRITE(*,199) (PLATFORM(J), J=LOOP*6+1,NUMPLAT(1))
        ENDIF
      ELSE
        COUNT = 0
        I = 1
        WRITE(*,188)
        DO 15 WHILE (COUNT .NE. NUMPLAT(PICK))
          IF (TYPE(I) .EQ. CODE(PICK)) THEN
            COUNT = COUNT + 1
            WRITE(*,177) PLATFORM(I),HRS(I),COST(I)
          ENDIF
          I = I + 1
15      ENDDO
      ENDIF
      ANS = 'A'
      DO 20 WHILE (ANS .NE. 'E')
        WRITE(*,55)
        READ(*,66) ANS
        IF (ANS .EQ. 'A') THEN
          IF (NUMPLAT(1) .EQ. 24) THEN
            WRITE(*,144)
          ELSE
            WRITE(*,77) 24 - NUMPLAT(1)
            WRITE(*,88)
            READ(*,99) TEMP
            LOCATE = FINDP(TEMP,PLATFORM,NUMPLAT)
            IF (LOCATE .EQ. 0) THEN
              PLATFORM(NUMPLAT(1)+1) = TEMP
              NUMPLAT(1) = NUMPLAT(1) + 1
              CALL PLATDATA(PLATFORM,NUMPLAT,TYPE,HRS,COST,QTY,CATA,
     +                      CATB,WORDA,WORDB,VALA,VALB,MISSIONS,
     +                      NUMISS,NUMDIST,DISTRICT)
            ELSE
              WRITE(*,133) TEMP
            ENDIF
          ENDIF
        ELSEIF (ANS .EQ. 'L') THEN
          WRITE(*,22) NUMPLAT(PICK), CG
          WRITE(*,188)
          IF (PICK .EQ. 1) THEN
            DO 50 I = 1, NUMPLAT(1)
              WRITE(*,177) PLATFORM(I),HRS(I),COST(I)
50          CONTINUE
          ELSE
            COUNT = 0
            I = 1
            DO 25 WHILE (COUNT .NE. NUMPLAT(PICK))
              IF (TYPE(I) .EQ. CODE(PICK)) THEN
                COUNT = COUNT + 1
```

```
              WRITE(*,177) PLATFORM(I),HRS(I),COST(I)
            ENDIF
            I = I + 1
25        ENDDO
        ENDIF
      ELSEIF (ANS .EQ. 'D') THEN
        IF (NUMPLAT(1) .LT. 2) THEN
          PRINT*,'*** ERROR ***'
          PRINT*,'AT LEAST 1 PLATFORM MUST REMAIN DEFINED'
        ELSE
          PRINT*,'ENTER PLATFORM NAME TO BE DELETED'
          READ(*,99) TEMP
          LOCATE = FINDP(TEMP,PLATFORM,NUMPLAT)
          IF (LOCATE .EQ. 0) THEN
            WRITE(*,111) TEMP
          ELSE
            DO 45 I = 2,4
              IF (TYPE(LOCATE) .EQ. CODE(I)) THEN
                NUMPLAT(I) = NUMPLAT(I) - 1
              ENDIF
45          CONTINUE
            IF (LOCATE .NE. NUMPLAT(1)) THEN
              DO 40 I = LOCATE, NUMPLAT(1)-1
                PLATFORM(I) = PLATFORM(I+1)
                TYPE(I) = TYPE(I+1)
                HRS(I) = HRS(I+1)
                COST(I) = COST(I+1)
                DO 41 J = 1,3
                  CATB(I,J) = CATB(I+1,J)
41              CONTINUE
                DO 42 J = 1,NUMISS
                  CATA(I,J) = CATA(I+1,J)
42              CONTINUE
40            CONTINUE
            ENDIF
            NUMPLAT(1) = NUMPLAT(1) - 1
          ENDIF
        ENDIF
      ELSEIF (ANS .EQ. 'C') THEN
        PRINT*,'ENTER NAME OF PLATFORM TO BE CHANGED'
        READ(*,99) TEMP
        LOCATE = FINDP(TEMP,PLATFORM,NUMPLAT)
        IF (LOCATE .EQ. 0) THEN
          WRITE(*,111) TEMP
        ELSE
          CALL CHGPLAT(NUMDIST,NUMPLAT,NUMISS,MISSIONS,DISTRICT,
     +                 PLATFORM,CATA,CATB,VALA,VALB,WORDA,WORDB,
     +                 HRS,QTY,COST,LOCATE,TEMP)
        ENDIF
      ELSEIF (ANS .EQ. 'E') THEN
        PRINT*,'EXITING EDIT PLATFORM ROUTINE'
      ELSE
```

```
          PRINT*,'*** ERROR ***'
          PRINT*,'PLEASE ENTER A,C,D,L OR E'
        ENDIF
20     ENDDO
11     FORMAT(I2)
22     FORMAT(' ','THERE ARE CURRENTLY ',I2,' PLATFORMS DEFINED',
     +       ' AS ',A26,':')
33     FORMAT(1X,A8)
44     FORMAT(' ',A8)
55     FORMAT('/',' THE FOLLOWING OPTIONS ARE AVAILABLE:'/,' (A)DD A',
     +    ' PLATFORM',/,' (D)ELETE A PLATFORM',/,' (C)HANGE A PLATFORM',
     +    ' NAME OR DATA',/,' (L)IST CURRENT PLATFORMS AND DATA',/,
     +    ' (E)XIT EDIT ROUTINE',/,' ENTER YOUR CHOICE (A,D,C,L,E):')
66     FORMAT(A1)
77     FORMAT(' ','THERE IS ROOM IN THE DATABASE FOR ',I2,' MORE ',
     +       'PLATFORM(S)')
88     FORMAT(' ','ENTER NEW PLATFORM NAME (NO MORE THAN 8 CHARACTERS)')
99     FORMAT(A7)
111    FORMAT(' ','*** ERROR ***',/,' PLATFORM NAME: ',A8,' NOT FOUND.')
133    FORMAT(' ','*** ERROR ***',/,' PLATFORM NAME: ',A8,' ALREADY',
     +       ' EXISTS.')
144    FORMAT(' ','*** ERROR ***',/,' PLATFORM DATABASE FULL.  ONLY',
     +       ' 24 PLATFORM TYPES ALLOWED')
155    FORMAT(' ','PLEASE CHOOSE WHICH PLATFORM CATEGORY TO EDIT:',/,
     +          ' (A) DISTRICT SURFACE',/,' (B) DISTRICT AIR',/,
     +          ' (C) AREA SURFACE',/,' (D) ALL',/,
     +          ' ENTER A, B, C, OR D:')
166    FORMAT(' ','*** ERROR ***',/,2X,A1,' IS NOT A VALID CHOICE',/)
177    FORMAT(1X,A8,1X,I4,1X,F5.2)
188    FORMAT(1X,'NAME',5X,'HRS',2X,'COST',/,1X,'----',5X,'---',
     +       2X,'----')
199    FORMAT(' ',6(A8,2X))
       RETURN
       END
       INTEGER FUNCTION FINDP(TEMP,PLATFORM,NUMPLAT)
*************************************************************************
*     FUNCTION:  SEARCH FOR OCCURANCE OF PLATFORM NAME IN DATABASE      *
*                                                                       *
*     AUTHOR    :  LT J.E. TOMKO, USCG                                  *
*                                                                       *
*     WRITTEN   :  19 MAY 1991                                          *
*                                                                       *
*     MODIFIED  :  20 MAY 1991                                          *
*************************************************************************
* PURPOSE:  CONDUCT A SEQUENTIAL SEARCH OF SMALL DATABASE (ONLY 8       *
* ENTRIES) TO FIND IF A PLATFORM NAME, CHARACTER STRING OF LENGTH 7,     *
* IS LOCATED WITHIN THE DATABASE.  PASS BACK A VALUE OF 0 IF NOT        *
* FOUND, OTHERWISE PASS BACK THE LOCATION OF THE NAME WITHIN THE        *
* ARRAY.                                                                *
*************************************************************************
       INTEGER NUMPLAT(4), I
       CHARACTER TEMP*8, PLATFORM(24)*8
```

```
      FINDP = 0
      DO 10 I=1,NUMPLAT(1)
        IF (TEMP .EQ. PLATFORM(I)) FINDP = I
10    CONTINUE
      RETURN
      END
      SUBROUTINE PLATDATA(PLATFORM,NUMPLAT,TYPE,HRS,COST,QTY,CATA,
     +                    CATB,WORDA,WORDB,VALA,VALB,MISSIONS,NUMISS,
     +                    NUMDIST,DISTRICT)
*************************************************************************
*     SUBROUTINE:  ADD ASSOCIATED DATA FOR NEW PLATFORM                 *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  06 JUNE 1991                                        *
*                                                                       *
*     MODIFIED   :  09 JULY 1991                                        *
*************************************************************************
* PURPOSE:  IF A NEW PLATFORM IS ADDED TO THE DATABASE, COLLECT THE     *
* REQUIRED DATA ASSOCIATED WITH THE PLATFORM.                           *
*************************************************************************
      INTEGER I,J, NUMPLAT(4), HRS(24), QTY(24,7), NUMDIST
      INTEGER CATA(24,10), CATB(24,3), CNT, NUMISS
      REAL COST(24), TCOST, THOUR, TVAL, VALA(4), VALB(4)
      CHARACTER ANS*1, TYPE(24)*3, PLATFORM(24)*8, DISTRICT(7)*10
      CHARACTER WORDA(4)*18, WORDB(4)*18, MISSIONS(10)*5
      LOGICAL WRONG
      WRONG = .TRUE.
      DO 10 WHILE (WRONG)
        WRITE(*,11) PLATFORM(NUMPLAT(1))
        READ(*,22) ANS
        IF (ANS .EQ. 'A') THEN
          TYPE(NUMPLAT(1)) = 'IDS'
          NUMPLAT(2) = NUMPLAT(2) + 1
          WRONG = .FALSE.
        ELSEIF (ANS .EQ. 'B') THEN
          TYPE(NUMPLAT(1)) = 'IDA'
          NUMPLAT(3) = NUMPLAT(3) + 1
          WRONG = .FALSE.
        ELSEIF (ANS .EQ. 'C') THEN
          TYPE(NUMPLAT(1)) = 'IAS'
          NUMPLAT(4) = NUMPLAT(4) + 1
          WRONG = .FALSE.
        ENDIF
10    ENDDO
      WRONG = .TRUE.
      DO 20 WHILE (WRONG)
        WRITE(*,33) PLATFORM(NUMPLAT(1))
        READ(*,*) THOUR
        IF ((THOUR .GE. 0.0) .AND. (THOUR .LE. 8760.0)) THEN
          HRS(NUMPLAT(1)) = INT(THOUR)
          WRONG = .FALSE.
```

```
        ENDIF
20    ENDDO
      WRONG = .TRUE.
      DO 30 WHILE (WRONG)
        WRITE(*,44) PLATFORM(NUMPLAT(1))
        READ(*,*) TCOST
        IF (TCOST .GE. 0.0) THEN
          COST(NUMPLAT(1)) = TCOST
          WRONG = .FALSE.
        ENDIF
30    ENDDO
      WRITE(*,55)
      WRITE(*,66)
      DO 40 I = 1,4
        WRITE(*,77) I, INT(VALB(I)*100.0), WORDB(I)
40    CONTINUE
      CNT = 0
      DO 45 WHILE (CNT .LT. 3)
        WRITE(*,99) CNT+1
        READ(*,*) TVAL
        IF ((INT(TVAL) .LT. 1) .OR. (INT(TVAL) .GT. 4)) THEN
         WRITE(*,111) INT(TVAL)
        ELSE
         CNT = CNT + 1
         CATB(NUMPLAT(1),CNT) = INT(TVAL)
        ENDIF
45    ENDDO
      WRITE(*,88)
      WRITE(*,66)
      DO 50 I = 1,4
        WRITE(*,77) I, INT(VALA(I)*100.0), WORDA(I)
50    CONTINUE
      CNT = 0
      DO 60 WHILE (CNT .LT. NUMISS)
        WRITE(*,122) MISSIONS(CNT+1)
        READ(*,*) TVAL
        IF ((INT(TVAL) .LT. 1) .OR. (INT(TVAL) .GT. 4)) THEN
         WRITE(*,111) INT(TVAL)
        ELSE
         CNT = CNT + 1
         CATA(NUMPLAT(1),CNT) = INT(TVAL)
        ENDIF
60    ENDDO
      DO 70 I = 1, NUMDIST
        WRONG = .TRUE.
        DO 80 WHILE (WRONG)
          WRITE(*,133) PLATFORM(NUMPLAT(1)), DISTRICT(I)
          READ(*,*) TQTY
          IF ((INT(TQTY) .LT. 0) .OR. (INT(TQTY) .GT. 100)) THEN
            WRITE(*,144)
          ELSE
           WRONG = .FALSE.
```

```
          QTY(NUMPLAT(1),I) = INT(TQTY)
         ENDIF
80      ENDDO
70    CONTINUE
11    FORMAT(' ','PLEASE SPECIFY THE PLATFORM ( ',A8,' ) AS EITHER',/,
     +       4X,'(A) DISTRICT SURFACE PLATFORM',/,4X,'(B) DISTRICT ',
     +       'AIR PLATFORM',/,4X,'(C) AREA SURFACE PLATFORM',/,
     +       ' ENTER EITHER A, B, OR C:')
22    FORMAT(A1)
33    FORMAT(' ','HOW MANY HOURS IS PLATFORM ( ',A8,' ) AVAILABLE',
     +       /,' FOR ASSIGNED MISSION OPERATIONS PER YEAR?',/,
     +       ' PLEASE ENTER A WHOLE NUMBER BETWEEN 0 AND 8760:')
44    FORMAT(' ','WHAT IS THE OPERATING COST PER YEAR FOR PLATFORM',
     +       ' ( ',A8,' )',/,' IN MILLIONS OF DOLLARS?',/,' PLEASE ',
     +       'ENTER A VALUE GREATER THAN OR EQUAL TO 0:')
55    FORMAT(/,' THE FOLLOWING VALUES AND DESCRIPTIONS ARE DEFINED ',
     +       'FOR',/,' CLASS CODES:')
66    FORMAT(' ',5X,'CODE',3X,'VALUE',8X,'DESCRIPTION',/,6X,4('-'),
     +       3X,5('-'),4X,18('-'))
77    FORMAT(' ',7X,I1,5X,I3,'%',5X,A18)
88    FORMAT(/,' THE FOLLOWING VALUES AND DESCRIPTIONS ARE DEFINED ',
     +       'FOR',/,' MISSION ASSIGNMENT CODES:')
99    FORMAT(/,' PLEASE ENTER A CLASS CODE (1-4) FOR LEVEL',
     +         1X,I1,' :')
111   FORMAT(' ','*** ERROR ***',/,' (',I1,') IS NOT A VALID ENTRY')
122   FORMAT(/,' PLEASE ENTER A MISSION ASSIGNMENT CODE (1-4) FOR ',
     +         'MISSION (',A5,'):')
133   FORMAT(/,' HOW MANY ',A8,' PLATFORMS ARE IN DISTRICT ',A10,
     +       ' ?',/,' PLEASE ENTER A NUMBER (0-100) :')
144   FORMAT(' ','*** ERROR ***',/,' INVALID ENTRY')
      RETURN
      END
      SUBROUTINE CHGPLAT(NUMDIST,NUMPLAT,NUMISS,MISSIONS,DISTRICT,
     +                   PLATFORM,CATA,CATB,VALA,VALB,WORDA,WORDB,
     +                   HRS,QTY,COST,LOCATE,TEMP)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  MODIFY NAME OR DATA FOR AN EXISTING PLATFORM         *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  30 AUGUST 1991                                      *
*                                                                       *
*     MODIFIED   :  30 AUGUST 1991                                      *
*************************************************************************
* PURPOSE:  CHANGE THE NAME OR ASSOCIATED DATA FOR A CURRENTLY          *
* DEFINED PLATFORM.                                                     *
*************************************************************************
      INTEGER I, J, NUMPLAT(4), NUMDIST, FINDP, LOCATE, START, END
      INTEGER NUMISS, HRS(24), QTY(24,7), PICK, HOLD
      INTEGER CATA(24,10), CATB(24,3)
      REAL THRS, TQTY, VALA(4), VALB(4), COST(24), TCOST
```

```
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER ANS*1, OPT*1, TEMP*8, WORDA(4)*18, WORDB(4)*18
      CHARACTER TEMP1*8
      LOGICAL ERROR, WRONG
      ANS = 'A'
      DO 10 WHILE (ANS .NE. 'E')
        WRITE(*,11) PLATFORM(LOCATE)
        READ(*,22) ANS
        IF (ANS .EQ. 'N') THEN
          WRITE(*,33) TEMP
          READ(*,44) TEMP1
          HOLD = FINDP(TEMP1,PLATFORM,NUMPLAT)
          IF (HOLD .EQ. 0) THEN
            PLATFORM(LOCATE) = TEMP1
            TEMP = TEMP1
          ELSE
            WRITE(*,55) TEMP1
          ENDIF
         ELSEIF (ANS .EQ. 'D') THEN
           OPT = 'A'
           DO 20 WHILE (OPT .NE. 'E')
             WRITE(*,77)
             READ(*,22) OPT
             IF (OPT .EQ. 'C') THEN
               PICK = 0
               DO 80 WHILE ((PICK .LT. 1) .OR. (PICK .GT. 5))
                 WRITE(*,188) TEMP, (I+2, I, I=1,3)
                 WRITE(*,144) 5
                 READ(*,*) PICK
                 IF ((PICK .LT. 1) .OR. (PICK .GT. 5)) THEN
                   WRITE(*,155)
                 ENDIF
80             ENDDO
               IF (PICK .GT. 1) THEN
                 IF (PICK .EQ. 2) THEN
                   START = 1
                   END = 3
                 ELSE
                   START = PICK-2
                   END = PICK-2
                 ENDIF
                 DO 90 I = START,END
                   ERROR = .TRUE.
                   DO 100 WHILE (ERROR)
                     WRITE(*,199) TEMP,I,WORDB(CATB(LOCATE,I))
     +                              (J, WORDB(J),VALB(J), J=1,4)
                     READ(*,*) TCAT
                     IF ((TCAT .LT. 1.0) .OR. (TCAT .GT. 4.0)) THEN
                       WRITE(*,155)
                     ELSE
                       ERROR = .FALSE.
                       CATB(LOCATE,I) = INT(TCAT)
```

```
                ENDIF
100           ENDDO
90          CONTINUE
          ENDIF
        ELSEIF (OPT .EQ. 'M') THEN
          PICK = 0
          DO 1'' WHILE ((PICK .LT. 1) .OR. (PICK .GT. NUMISS+2))
            WRITE(*,222) TEMP, (I+2, MISSIONS(I), I=1,NUMISS)
            WRITE(*,144) NUMISS+2
            READ(*,*) PICK
            IF ((PICK .LT. 1) .OR. (PICK .GT. NUMISS+2)) THEN
              WRITE(*,155)
            ENDIF
110         ENDDO
            IF (PICK .GT. 1) THEN
              IF (PICK .EQ. 2) THEN
                START = 1
                END = NUMISS
              ELSE
                START = PICK-2
                END = PICK-2
              ENDIF
              DO 120 I = START,END
                ERROR = .TRUE.
                DO 130 WHILE (ERROR)
                  WRITE(*,211) TEMP, MISSIONS(I),
     +                         WORDA(CATA(LOCATE,I)),
     +                         (J, WORDA(J),VALA(J), J=1,4)
                  READ(*,*) TCAT
                  IF ((TCAT .LT. 1.0) .OR. (TCAT .GT. 4.0)) THEN
                    WRITE(*,155)
                  ELSE
                    ERROR = .FALSE.
                    CATA(LOCATE,I) = INT(TCAT)
                  ENDIF
130               ENDDO
120           CONTINUE
            ENDIF
          ELSEIF (OPT .EQ. 'H') THEN
            ERROR = .TRUE.
            DO 30 WHILE (ERROR)
              WRITE(*,88) TEMP, HRS(LOCATE)
              READ(*,*) THRS
              IF ((THRS .LT. 0.0) .OR. (THRS .GT. 8760.0)) THEN
                WRITE(*,99)
              ELSE
                ERROR = .FALSE.
                HRS(LOCATE) = INT(THRS)
              ENDIF
30          ENDDO
          ELSEIF (OPT .EQ. 'O') THEN
            ERROR = .TRUE.
```

```
              DO 40 WHILE (ERROR)
                WRITE(*,111) TEMP, INT(COST(LOCATE)*1000.0)
                READ(*,*) TCOST
                IF (TCOST .LT. 0.0) THEN
                  WRITE(*,122)
                ELSE
                  ERROR = .FALSE.
                  COST(LOCATE) = TCOST/1000.0
                ENDIF
40            ENDDO
            ELSEIF (OPT .EQ. 'N') THEN
              PICK = 0
              DO 50 WHILE ((PICK .LT. 1) .OR.
     +                     (PICK .GT. NUMDIST+2))
                WRITE(*,133) TEMP, (I+2, DISTRICT(I), I=1,NUMDIST)
                WRITE(*,144) NUMDIST+2
                READ(*,*) PICK
                IF ((PICK .LT. 1) .OR. (PICK .GT. NUMDIST+2)) THEN
                  WRITE(*,155)
                ENDIF
50            ENDDO
              IF (PICK .GT. 1) THEN
                IF (PICK .EQ. 2) THEN
                  START = 1
                  END = NUMDIST
                ELSE
                  START = PICK-2
                  END = PICK-2
                ENDIF
                DO 60 I = START,END
                  ERROR = .TRUE.
                  DO 70 WHILE (ERROR)
                    WRITE(*,166) QTY(LOCATE,I),TEMP,DISTRICT(I)
                    READ(*,*) TQTY
                    IF (TQTY .LT. 0.0) THEN
                      WRITE(*,177)
                    ELSE
                      ERROR = .FALSE.
                      QTY(LOCATE,I) = INT(TQTY)
                    ENDIF
70                ENDDO
60              CONTINUE
              ENDIF
            ELSEIF (OPT .NE. 'E') THEN
              WRITE(*,66) OPT
            ENDIF
20        ENDDO
        ELSEIF (ANS .NE. 'E') THEN
          WRITE(*,66) ANS
        ENDIF
10    ENDDO
11    FORMAT(' ','FOR PLATFORM ( ',A8,' ) YOU CAN CHANGE EITHER:',/,
```

```
     +       ' (N)AME OF PLATFORM',/,' (D)ATA CONCERNING THE PLATFORM',
     +      /,' (E)XIT THIS ROUTINE',/,' ENTER YOUR CHOICE (N,D,E):')
22    FORMAT(A1)
33    FORMAT(' ','ENTER NEW NAME FOR PLATFORM (',A8,').')
44    FORMAT(A8)
55    FORMAT(' ','*** ERROR ***',/,' PLATFORM NAME: ',A8,' ALREADY',
     +       ' EXISTS.')
66    FORMAT(' ','*** ERROR ***',/,' (',A1,') IS NOT A VALID ENTRY.',/)
77    FORMAT(' ','YOU CAN CHANGE THE FOLLOWING PLATFORM DATA:',/,
     +       ' (C)APABILITY RATINGS',/,' (M)ISSION RATINGS',/,
     +       ' (O)PERATING COST',/,' (H)OURS AVAILABLE',/,' (N)UMBER',
     +       ' AND LOCATION',/,' (E)XIT THIS ROUTINE',/,' ENTER ',
     +       'YOUR CHOICE (C,M,O,H,N,E):')
88    FORMAT(' ',A8,' CURRENTLY HAS ',I4,' OPERATIONAL HOURS ',
     +       'AVAILABLE',/,' ENTER THE NEW NUMBER OF HOURS ',
     +       '(BETWEEN 0 AND 8760):')
99    FORMAT(' ','*** ERROR ***',/,' TOTAL HOURS AVAILABLE MUST BE ',
     +       'BETWEEN 0 AND 8760',/)
111   FORMAT(' ',A8,' CURRENTLY COST $',I4,' THOUSAND TO OPERATE PER',
     +       ' YEAR',/,' ENTER THE REVISED OPERATING COST IN THOUSANDS'
     +       ,/,' ROUNDED TO THE NEAREST THOUSAND. (1 MILLION = 1000):')
122   FORMAT(' ','*** ERROR ***',/,' ANNUAL COST CANNOT BE NEGATIVE',/)
133   FORMAT(' ','YOU CAN CHANGE THE NUMBER OF ',A8,' IN THE FOLLOWING'
     +            ,' DISTRICTS:',/,'  1) EXIT',8X,'2) ALL',9X,
     +             3(1X,I2,') ',A10),/,4(1X,I2,') ',A10))
144   FORMAT(' ','ENTER YOUR CHOICE FOR MODIFICATION (1 - ',I2,'):')
155   FORMAT(' ','*** ERROR ***',/,' INVALID ENTRY.',/)
166   FORMAT(' ','THERE ARE CURRENTLY ',I2,1X,A8,' PLATFORMS IN ',
     +            'DISTRICT ( ',A10,' ).',/,' ENTER THE NEW NUMBER OF',
     +            ' PLATFORMS LOCATED WITHIN THE DISTRICT:')
177   FORMAT(' ','*** ERROR ***',/,' NUMBER OF PLATFORMS CANNOT BE',
     +            ' NEGATIVE.',/)
188   FORMAT(' ','YOU CAN CHANGE RATING OF ',A8,' FOR THE FOLLOWING'
     +            ,' CLASSES:',/,'  1) EXIT',4X,'2) ALL',2X,
     +             3(3X,I2,') ',I2,2X))
199   FORMAT(' ','THE CURRENT RATING FOR PLATFORM TYPE: ',A8,/,
     +            ' FOR CLASS (',I1,') IS ',A18,/,' THE ',
     +            'FOLLOWING RATINGS ARE AVAILABLE:',/,' RATING',2X,
     +            'DESCRIPTION',11X,'VALUE',/,1X,6('-'),2X,17('-'),5X,
     +            5('-'),4(/,3X,I1,5X,A18,4X,F4.2),/,' ENTER NEW ',
     +            'RATING (1-4):')
211   FORMAT(' ','THE CURRENT RATING FOR PLATFORM TYPE: ',A8,/,
     +            ' FOR MISSION ( ',A5,' ) IS ',A18,/,' THE ',
     +            'FOLLOWING RATINGS ARE AVAILABLE:',/,' RATING',2X,
     +            'DESCRIPTION',11X,'VALUE',/,1X,6('-'),2X,17('-'),5X,
     +            5('-'),4(/,3X,I1,5X,A18,4X,F4.2),/,' ENTER NEW ',
     +            'RATING (1-4):')
222   FORMAT(' ','YOU CAN CHANGE THE RATING FOR ',A8,' IN THE ',
     +            'FOLLOWING MISSIONS:',/,'  1) EXIT',3X,'2) ALL',2X,
     +            3(1X,I2,') ',A5),/,2(5(1X,I2,') ',A5)))
      RETURN
      END
```

```
      SUBROUTINE EDITDIST(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,QTY,
     +                    MISSIONS,DISTRICT,PLATFORM,TYPE,AREA,INDP,
     +                    LOCINDP)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  EDIT DISTRICT LISTING                                *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  18 MAY 1991                                         *
*                                                                       *
*     MODIFIED   :  11 JUNE 1991                                        *
*************************************************************************
* PURPOSE:  MENU DRIVEN USER INTERFACE TO ALLOW FOR MODIFICATION OF      *
* THE COAST GUARD DISTRICT DATA. USER MAY ADD, DELETE, CHANGE, AND       *
* LIST THE CURRENT DISTRICTS ON FILE. THE MAXIMUM NUMBER OF DISTRICTS    *
* WAS SET AT 7/AREA.  THIS IS TO AVOID EXTREMELY LARGE GAMS MODELS.      *
* ROUTINE IS WRITTEN WITH ERROR CHECKING CODE TO AVOID DUPLICATION       *
* OF DISTRICT NAMES OR ATTEMPTING DELETION OF NONEXISTING NAMES.         *
*************************************************************************
      INTEGER I, J, K, NUMPLAT(4), LOCATE, FINDD, HOLD, NUMDIST
      INTEGER NUMISS, HRS(24), HOURS(7,10,6), QTY(24,7)
      INTEGER CATB(24,3), CATA(24,10), INDP, LOCINDP(7)
      REAL COST(24)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, TEMP*10
      CHARACTER PLATFORM(24)*8, ANS*1, TYPE(24)*3, AREA*8
      ANS = 'A'
      DO 10 WHILE (ANS .NE. 'E')
        WRITE (*,22) NUMDIST, AREA
        IF (NUMDIST .LT. 5) THEN
          WRITE(*,33) (DISTRICT(I), I=1,NUMDIST)
        ELSE
          WRITE(*,33) (DISTRICT(I), I=1,4)
          WRITE(*,33) (DISTRICT(I), I=5,NUMDIST)
        ENDIF
        WRITE(*,55)
        READ(*,66) ANS
        IF (ANS .EQ. 'A') THEN
          IF (NUMDIST .EQ. 7) THEN
            WRITE(*,144)
          ELSE
            WRITE(*,77) 7-NUMDIST
            WRITE(*,88)
            READ(*,99) TEMP
            LOCATE = FINDD(TEMP,DISTRICT,NUMDIST)
            IF (LOCATE .EQ. 0) THEN
              DISTRICT(NUMDIST+1) = TEMP
              NUMDIST = NUMDIST + 1
              CALL ADDDDATA(NUMDIST,DISTRICT,MISSIONS,NUMISS,HOURS,QTY,
     +                      NUMPLAT,PLATFORM,ANS,INDP,LOCINDP,LOCATE)
            ELSE
              WRITE(*,133) TEMP
```

```
          ENDIF
        ENDIF
      ELSEIF (ANS .EQ. 'L') THEN
        WRITE (*,22) NUMDIST, AREA
        DO 20 I = 1, NUMDIST
          WRITE(*,*) DISTRICT(I)
20      CONTINUE
      ELSEIF (ANS .EQ. 'D') THEN
        IF (NUMDIST .LT. 2) THEN
          PRINT*,'*** ERROR ***'
          PRINT*,'AT LEAST 1 DISTRICT MUST REMAIN DEFINED'
        ELSE
          PRINT*,'ENTER DISTRICT NAME TO BE DELETED'
          READ(*,99) TEMP
          LOCATE = FINDD(TEMP,DISTRICT,NUMDIST)
          IF (LOCATE .EQ. 0) THEN
            WRITE(*,111) TEMP
          ELSE
            IF (LOCATE .NE. NUMDIST) THEN
              DO 30 I = LOCATE, NUMDIST-1
                DISTRICT(I) = DISTRICT(I+1)
30            CONTINUE
            ENDIF
            NUMDIST = NUMDIST - 1
          ENDIF
        ENDIF
      ELSEIF (ANS .EQ. 'C') THEN
        PRINT*,'ENTER NAME OF THE DISTRICT TO BE MODIFIED'
        READ(*,99) TEMP
        LOCATE = FINDD(TEMP,DISTRICT,NUMDIST)
        IF (LOCATE .EQ. 0) THEN
          WRITE(*,111) TEMP
        ELSE
          CALL CHGDDATA(NUMDIST,DISTRICT,MISSIONS,NUMISS,HOURS,QTY,
     +                  NUMPLAT,PLATFORM,ANS,INDP,LOCINDP,LOCATE)
        ENDIF
      ELSEIF (ANS .EQ. 'E') THEN
        PRINT*,'EXITING EDIT DISTRICT ROUTINE'
      ELSE
        PRINT*,'*** ERROR ***'
        PRINT*,'PLEASE ENTER A,C,D,L OR E'
      ENDIF
10    ENDDO
11    FORMAT(I2)
22    FORMAT(' ','THERE ARE CURRENTLY ',I2,' DISTRICTS DEFINED',
     +       ' FOR ',A8,' AREA:')
33    FORMAT(3X,4(A10,5X))
55    FORMAT('/',' THE FOLLOWING OPTIONS ARE AVAILABLE:'/,' (A)DD A',
     +   ' DISTRICT',/,' (D)ELETE A DISTRICT',/,' (C)HANGE A DISTRICT',
     +   ' NAME OR DATA',/,' (L)IST CURRENT DISTRICTS',/,' (E)XIT EDIT'
     +   ,' ROUTINE',/,' ENTER YOUR CHOICE (A,D,C,L,E):')
66    FORMAT(A1)
```

```
77    FORMAT(' ','THERE IS ROOM IN THE DATABASE FOR ',I2,' MORE ',
     +       'DISTRICT(S)')
88    FORMAT(' ','ENTER NEW DISTRICT NAME (NO MORE THAN 10 CHARACTERS)')
99    FORMAT(A10)
111   FORMAT(' ','*** ERROR ***',/,' DISTRICT NAME: ',A10,' NOT FOUND.')
133   FORMAT(' ','*** ERROR ***',/,' DISTRICT NAME: ',A10,' ALREADY',
     +       ' EXISTS.')
144   FORMAT(' ','*** ERROR ***',/,' DISTRICT DATABASE FULL.  ONLY',
     +       ' 7 DISTRICTS ALLOWED FOR EACH AREA.')
      RETURN
      END
      INTEGER FUNCTION FINDD(TEMP,DISTRICT,NUMDIST)
*************************************************************************
*     FUNCTION:  SEARCH FOR OCCURANCE OF DISTRICT NAME IN DATABASE      *
*                                                                       *
*     AUTHOR    :  LT J.E. TOMKO, USCG                                  *
*                                                                       *
*     WRITTEN   :  18 MAY 1991                                          *
*                                                                       *
*     MODIFIED  :  04 JUNE 1991                                         *
*************************************************************************
* PURPOSE:  CONDUCT A SEQUENTIAL SEARCH OF SMALL DATABASE (ONLY 7       *
* ENTRIES) TO FIND IF A DISTRICT NAME, CHARACTER STRING OF LENGTH 11,   *
* IS LOCATED WITHIN THE DATABASE.  PASS BACK A VALUE OF 0 IF NOT        *
* FOUND, OTHERWISE PASS BACK THE LOCATION OF THE NAME WITHIN THE        *
* ARRAY.                                                                *
*************************************************************************
      INTEGER I, NUMDIST
      CHARACTER TEMP*10, DISTRICT(7)*10
      FINDD = 0
      DO 10 I=1, NUMDIST
        IF (TEMP .EQ. DISTRICT(I)) FINDD = I
10    CONTINUE
      RETURN
      END
      SUBROUTINE CHGDDATA(NUMDIST,DISTRICT,MISSIONS,NUMISS,HOURS,QTY,
     +                    NUMPLAT,PLATFORM,ANS,INDP,LOCINDP,LOCATE)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  MODIFY NAME OR DATA FOR AN EXISTING DISTRICT         *
*                                                                       *
*     AUTHOR    :  LT J.E. TOMKO, USCG                                  *
*                                                                       *
*     WRITTEN   :  18 MAY 1991                                          *
*                                                                       *
*     MODIFIED  :  14 JUNE 1991                                         *
*************************************************************************
* PURPOSE:  CHANGE THE NAME OR ASSOCIATED DATA FOR A CURRENTLY          *
* DEFINED DISTRICT.                                                     *
*************************************************************************
      INTEGER I, J, NUMPLAT(4), NUMDIST, INDP, LOCINDP(7), FOUND
      INTEGER NUMISS, HOURS(7,10,6), QTY(24,7), LOCATE, FINDD, PICK
```

```
      INTEGER START, END
      REAL THOUR(3), TQTY
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER ANS*1, OPT*1, TEMP*10, STR(2)*7
      LOGICAL ERROR, WRONG
      DATA STR/'SURFACE','AIR    '/
      ERROR = .TRUE.
      DO 10 WHILE (ERROR)
        WRITE(*,11) DISTRICT(LOCATE)
        READ(*,22) OPT
        IF (OPT .EQ. 'N') THEN
          WRITE(*,122) DISTRICT(LOCATE)
          HOLD = LOCATE
          READ(*,99) TEMP
          LOCATE = FINDD(TEMP,DISTRICT,NUMDIST)
          IF (LOCATE .EQ. 0) THEN
            DISTRICT(HOLD) = TEMP
            ERROR = .FALSE.
          ELSE
            WRITE(*,133) TEMP
          ENDIF
        ELSEIF ( OPT .EQ. 'D') THEN
          DO 50 WHILE (ERROR)
            WRITE(*,77)
            READ(*,22) OPT
            IF (OPT .EQ. 'P') THEN
              PICK = 0
              DO 90 WHILE (PICK .NE. 1)
                PICK = 0
                DO 20 WHILE ((PICK .LT. 1) .OR.
     +                      (PICK .GT. NUMPLAT(1)+2))
                  WRITE(*,155) (I+2, PLATFORM(I), I=1,NUMPLAT(1))
                  WRITE(*,166) NUMPLAT(1)+2
                  READ(*,*) PICK
                  IF ((PICK .LT. 1) .OR. (PICK .GT. NUMPLAT(1)+2)) THEN
                    WRITE(*,177) PICK
                  ENDIF
20                ENDDO
                  IF (PICK .GT. 1) THEN
                    IF (PICK .EQ. 2) THEN
                      START = 1
                      END = NUMPLAT(1)
                    ELSE
                      START = PICK-2
                      END = PICK-2
                    ENDIF
                    DO 30 I = START, END
                      WRONG = .TRUE.
                      DO 40 WHILE (WRONG)
                        WRITE(*,188) QTY(I,LOCATE), PLATFORM(I),
     +                               DISTRICT(LOCATE)
                        READ(*,*) TQTY
```

```
                    IF (TQTY .LT. 0.0) THEN
                      WRITE(*,199)
                    ELSE
                      WRITE(*,199)
                      QTY(I,LOCATE) = INT(TQTY)
                    ENDIF
40                ENDDO
30              CONTINUE
              ENDIF
90          ENDDO
          ELSEIF (OPT .EQ. 'R') THEN
            PICK = 0
            DO 100 WHILE (PICK .NE. 1)
              PICK = 0
              DO 110 WHILE ((PICK .LT. 1) .OR.
     +                      (PICK .GT. NUMISS+2))
                WRITE(*,44) (I+2, MISSIONS(I), I=1,NUMISS)
                WRITE(*,166) NUMISS+2
                READ(*,*) PICK
                IF ((PICK .LT. 1) .OR. (PICK .GT. NUMISS+2)) THEN
                  WRITE(*,177) PICK
                ENDIF
110             ENDDO
                IF (PICK .GT. 1) THEN
                  IF (PICK .EQ. 2) THEN
                    START = 1
                    END = NUMPLAT(1)
                  ELSE
                    START = PICK-2
                    END = PICK-2
                  ENDIF
                  DO 120 I = START, END
                    DO 150 K = 1,2
                      WRONG = .TRUE.
                      DO 130 WHILE (WRONG)
                        WRITE(*,55) DISTRICT(LOCATE),STR(K),MISSIONS(I)
     +                             ,(HOURS(LOCATE,I,J+(K-1)*3),J=1,3)
     +                             , STR(K)
                        READ(*,*) (THOUR(J), J=1,3)
                        IF ((THOUR(1) .LT. 0.0)
     +                       .OR. (THOUR(2) .LT. 0.0)
     +                       .OR. (THOUR(3) .LT. 0.0)) THEN
                          WRITE(*,199)
                        ELSE
                          WRONG = .FALSE.
                          DO 140 J = 1,3
                            HOURS(LOCATE,I,J+(K-1)*3) = INT(THOUR(J))
140                       CONTINUE
                        ENDIF
130                   ENDDO
150                 CONTINUE
120               CONTINUE
```

```
              ENDIF
100           ENDDO
            ELSEIF (OPT .NE. 'E') THEN
              WRITE(*,33) OPT
            ELSE
              ERROR = .FALSE.
            ENDIF
50        ENDDO
          FOUND = 0
          DO 60 I=1,INDP
            IF (LOCINDP(I) .EQ. LOCATE) FOUND = I
60        CONTINUE
          DO 70 WHILE ((OPT .NE. 'Y') .AND. (OPT .NE. 'N'))
            WRITE(*,144) DISTRICT(LOCATE)
            READ(*,22) OPT
            IF ((OPT .NE. 'Y') .AND. (OPT .NE. 'N')) THEN
              WRITE(*,33) OPT
            ENDIF
70        ENDDO
          IF ((OPT .EQ. 'N') .AND. (FOUND .EQ. 0)) THEN
            INDP = INDP + 1
            LOCINDP(INDP) = LOCATE
          ELSEIF ((OPT .EQ. 'Y') .AND. (FOUND .GT. 0)) THEN
            DO 80 I = FOUND, INDP-1
              LOCINDP(I) = LOCINDP(I+1)
80          CONTINUE
            INDP = INDP - 1
          ENDIF
        ELSE
          WRITE(*,33) OPT
        ENDIF
10     ENDDO
11     FORMAT(' ','FOR DISTRICT ( ',A10,' ) YOU CAN CHANGE EITHER:',
     +       /,' (N)AME OF DISTRICT',/,' (D)ATA CONCERNING THE ',
     +       'DISTRICT',/,' ENTER YOUR CHOICE (N OR D):')
22     FORMAT(A1)
33     FORMAT(' ','*** ERROR ***',/,' (',A1,') IS NOT A VALID ENTRY',/)
44     FORMAT(' ','YOU CAN CHANGE THE HOURS NEEDED FOR THE FOLLOWING ',
     +       'MISSIONS:',/,'  1) EXIT   2) ALL',2X,3(1X,I2,') ',A5)
     +       ,/,2(5(1X,I2,') ',A5),/))
55     FORMAT(' ','FOR DISTRICT ( ',A10,' ) THE CURRENT ',A7,1X,A5,/,
     +       ' MISSION RESOURCE HOUR REQUIREMENTS EXIST:',3(I7,1X),/,
     +       ' ENTER THE NEW NUMBER OF ',A7,/,' RESOURCE HOURS ',
     +       'NEEDED FOR EACH CLASS (1,2,3):',/,
     +       ' ENTER 3 WHOLE NUMBERS SEPERATED BY COMMAS:')
66     FORMAT(' ','*** ERROR ***',/,' NEGATIVE VALUES NOT ALLOWED.',/)
77     FORMAT(' ','YOU CAN MODIFY THE FOLLOWING DISTRICT DATA:',/,
     +       ' (P)LATFORM QUANTITIES',/,' (R)ESOURCE HOURS REQUIRED',/,
     +       ' (E)XIT THIS MENU',/,' ENTER CHOICE (P,R,E):')
88     FORMAT(/,' NUMBER AND TYPES OF PLATFORMS LOCATED IN DISTRICT:')
99     FORMAT(A10)
122    FORMAT(' ','ENTER NEW NAME FOR DISTRICT (',A10,').')
```

```
133   FORMAT(' ','*** ERROR ***',/,' DISTRICT NAME: ',A10,' ALREADY',
     +       ' EXISTS.')
144   FORMAT(' ','CAN AREA SURFACE PLATFORMS OPERATE IN DISTRICT ( ',
     +          A10,' )?',/,' ENTER (Y OR N):')
155   FORMAT(' ','YOU CAN CHANGE THE QUANTITY OF THE FOLLOWING ',
     +       'PLATFORMS:',/,'  1) EXIT      2) ALL',5X,3(1X,I2,') ',A8)
     +       ,/,5(5(1X,I2,') ',A8),/))
166   FORMAT(' ','ENTER YOUR CHOICE FOR MODIFICATION (1 -',I2,'):')
177   FORMAT(' ','*** ERROR ***',/,' (',I9,' ) IS NOT A VALID ENTRY',/)
188   FORMAT(' ','THERE ARE CURRENTLY ',I2,' ( ',A8,' ) PLATFORMS',
     +          /,' LOCATED IN DISTRICT ( ',A10,' ).',/,
     +          ' ENTER THE NEW NUMBER OF PLATFORMS:')
199   FORMAT(' ','*** ERROR ***',/,' NEGATIVE NUMBERS NOT ALLOWED',/)
      RETURN
      END
      SUBROUTINE ADDDDATA(NUMDIST,DISTRICT,MISSIONS,NUMISS,HOURS,QTY,
     +                    NUMPLAT,PLATFORM,ANS,INDP,LOCINDP,LOCATE)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  ADD DISTRICT DATA                                    *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  14 JUNE 1991                                        *
*                                                                       *
*     MODIFIED   :  14 JUNE 1991                                        *
*************************************************************************
* PURPOSE:  COLLECT NEEDED DATA FOR AN ADDITIONAL DISTRICT.             *
*************************************************************************
      INTEGER I, J, NUMPLAT(4), NUMDIST, INDP, LOCINDP(7), FOUND
      INTEGER NUMISS, HOURS(7,10,6), QTY(24,7), LOCATE, FINDD
      REAL THOUR(3), TQTY
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER ANS*1, OPT*1, TEMP*10, STR(2)*7
      LOGICAL ERROR
      DATA STR/'SURFACE','AIR    '/
      ERROR = .TRUE.
      WRITE(*,22) DISTRICT(NUMDIST)
      DO 10 I = 1, NUMISS
        DO 20 K = 1,2
          ERROR = .TRUE.
          DO 30 WHILE (ERROR)
            WRITE(*,55) MISSIONS(I), STR(K)
            READ(*,*) (THOUR(J), J=1,3)
            IF ((THOUR(1) .LT. 0) .OR. (THOUR(2) .LT. 0) .OR.
     +          (THOUR(3) .LT. 0)) THEN
              WRITE(*,66)
            ELSE
              DO 40 J = 1,3
                HOURS(NUMDIST,I,J+(K-1)*3) = INT(THOUR(J))
40            CONTINUE
              ERROR = .FALSE.
```

```
            ENDIF
30        ENDDO
20      CONTINUE
10    CONTINUE
      WRITE(*,88)
      DO 50 I = 1, NUMPLAT(1)
        ERROR = .TRUE.
        DO 60 WHILE (ERROR)
          WRITE(*,44) PLATFORM(I)
          READ(*,*) TQTY
          IF (TQTY .LT. 0.0) THEN
            WRITE(*,66)
          ELSE
            QTY(I,NUMDIST) = INT(TQTY)
            ERROR = .FALSE.
          ENDIF
60      ENDDO
50    CONTINUE
      DO 90 WHILE ((OPT .NE. 'Y') .AND. (OPT .NE. 'N'))
        WRITE(*,77) DISTRICT(NUMDIST)
        READ(*,11) OPT
        IF ((OPT .NE. 'Y') .AND. (OPT .NE. 'N')) THEN
          WRITE(*,33) OPT
        ENDIF
90    ENDDO
      IF (OPT .EQ. 'N') THEN
        INDP = INDP + 1
        LOCINDP(INDP) = NUMDIST
      ENDIF
11    FORMAT(A1)
22    FORMAT(' ','FOR DISTRICT ( ',A10,' ) ENTER THE FOLLOWING DATA:',
     +       //,' MISSION RESOURCE HOUR REQUIREMENTS (SURFACE & AIR):')
33    FORMAT(' ','*** ERROR ***',/,' (',A1,') IS NOT A VALID ENTRY',/)
44    FORMAT(' ','HOW MANY (',A8,') PLATFORMS IN THIS DISTRICT:')
55    FORMAT(/,' FOR MISSION ( ',A5,' ) ENTER THE NUMBER OF ',A7,
     +       /,' RESOURCE HOURS NEEDED FOR EACH CLASS (1,2,3):',
     +       /,' ENTER 3 WHOLE NUMBERS SEPERATED BY COMMAS:')
66    FORMAT(' ','*** ERROR ***',/,' NEGATIVE VALUES NOT ALLOWED.',/)
77    FORMAT(' ','CAN AREA SURFACE PLATFORMS OPERATE IN DISTRICT ( ',
     +          A10,' )?',/,' ENTER (Y OR N):')
88    FORMAT(/,' NUMBER AND TYPES OF PLATFORMS LOCATED IN DISTRICT:')
99    FORMAT(A10)
      RETURN
      END
      SUBROUTINE EDITMISS(NUMDIST,NUMPLAT,NUMISS,HOURS,MISSIONS,
     +                    DISTRICT,PLATFORM,ATOS,STOA,CATA,VALA,WORDA)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  EDIT MISSION LISTING                                 *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
```

```
*     WRITTEN   :  18 MAY 1991                                          *
*                                                                       *
*     MODIFIED  :  16 JUNE 1991                                         *
*************************************************************************
* PURPOSE:  MENU DRIVEN USER INTERFACE TO ALLOW FOR MODIFICATION OF     *
* THE COAST GUARD MISSION DATA.  USER MAY ADD, DELETE, CHANGE, AND      *
* LIST THE CURRENT MISSIONS ON FILE.  THE MAXIMUM NUMBER OF MISSIONS    *
* WAS SET AT 10.  THIS IS TO AVOID EXTREMELY LARGE GAMS MODELS.  THIS   *
* ROUTINE IS WRITTEN WITH ERROR CHECKING CODE TO AVOID DUPLICATION      *
* OF MISSION NAMES OR ATTEMPTING DELETION OF NONEXISTING NAMES.         *
*************************************************************************
      INTEGER I, J, K, NUMPLAT(4), LOCATE, FIND, HOLD, NUMDIST
      INTEGER NUMISS, HRS(24), HOURS(7,10,6), QTY(24,7)
      INTEGER CATA(24,10)
      REAL COST(24), VALA(4)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, TEMP*5
      CHARACTER PLATFORM(24)*8, ANS*1, TYPE(24)*3, WORDA(4)*18
      ANS = 'A'
      DO 10 WHILE (ANS .NE. 'E')
        WRITE(*,22) NUMISS
        WRITE(*,155) (MISSIONS(I), I=1,NUMISS)
        WRITE(*,55)
        READ(*,66) ANS
        IF (ANS .EQ. 'A') THEN
          IF (NUMISS .EQ. 10) THEN
            WRITE(*,144)
          ELSE
            WRITE(*,77) 10-NUMISS
            WRITE(*,88)
            READ(*,99) TEMP
            LOCATE = FIND(TEMP,MISSIONS,NUMISS)
            IF (LOCATE .EQ. 0) THEN
              MISSIONS(NUMISS+1) = TEMP
              NUMISS = NUMISS + 1
              CALL ADDMISS(NUMDIST,NUMPLAT,NUMISS,HOURS,MISSIONS,
     +                     DISTRICT,PLATFORM,ATOS,STOA,CATA,VALA,WORDA)
            ELSE
              WRITE(*,133) TEMP
            ENDIF
          ENDIF
        ELSEIF (ANS .EQ. 'D') THEN
          IF (NUMISS .LT. 2) THEN
            PRINT*,'*** ERROR ***'
            PRINT*,'AT LEAST 1 MISSION MUST REMAIN DEFINED'
          ELSE
            PRINT*,'ENTER MISSION NAME TO BE DELETED'
            READ(*,99) TEMP
            LOCATE = FIND(TEMP,MISSIONS,NUMISS)
            IF (LOCATE .EQ. 0) THEN
              WRITE(*,111) TEMP
            ELSE
              IF (LOCATE .NE. NUMISS) THEN
```

```
                DO 20 I = LOCATE, NUMISS-1
                  MISSIONS(I) = MISSIONS(I+1)
20              CONTINUE
              ENDIF
              NUMISS = NUMISS - 1
            ENDIF
          ENDIF
        ELSEIF (ANS .EQ. 'C') THEN
          PRINT*,'ENTER MISSION NAME TO BE CHANGED'
          READ(*,99) TEMP
          LOCATE = FIND(TEMP,MISSIONS,NUMISS)
          IF (LOCATE .EQ. 0) THEN
            WRITE(*,111) TEMP
          ELSE
            CALL CHGMISS(NUMDIST,NUMPLAT,NUMISS,HOURS,MISSIONS,
     +                   DISTRICT,PLATFORM,ATOS,STOA,CATA,VALA,
     +                   WORDA,LOCATE,TEMP)
          ENDIF
        ELSEIF (ANS .EQ. 'E') THEN
          PRINT*,'EXITING EDIT MISSION ROUTINE'
        ELSE
          PRINT*,'*** ERROR ***'
          PRINT*,'PLEASE ENTER A,C,D OR E'
        ENDIF
10      ENDDO
11      FORMAT(I2)
22      FORMAT(' ','THERE ARE CURRENTLY ',I2,' MISSIONS DEFINED: ')
33      FORMAT(1X,A5)
44      FORMAT(' ',A5)
55      FORMAT('/',' THE FOLLOWING OPTIONS ARE AVAILABLE:'/,' (A)DD A',
     +       ' MISSION',/,' (D)ELETE A MISSION',/,' (C)HANGE A MISSION',
     +       ' NAME OR DATA',/,' (E)XIT EDIT ROUTINE',/,
     +       ' ENTER YOUR CHOICE (A,D,C,E):')
66      FORMAT(A1)
77      FORMAT(' ','THERE IS ROOM IN THE DATABASE FOR ',I2,' MORE ',
     +        'MISSION(S)')
88      FORMAT(' ','ENTER NEW MISSION NAME (NO MORE THAN 5 CHARACTERS)')
99      FORMAT(A5)
111     FORMAT(' ','*** ERROR ***',/,' MISSION NAME: ',A5,' NOT FOUND.')
122     FORMAT(' ','ENTER NEW NAME FOR MISSION (',A5,').')
133     FORMAT(' ','*** ERROR ***',/,' MISSION NAME: ',A5,' ALREADY',
     +        ' EXISTS.')
144     FORMAT(' ','*** ERROR ***',/,' MISSION DATABASE FULL.  ONLY 10',
     +        ' MISSIONS ALLOWED.')
155     FORMAT(2X,10(A5,2X))
        RETURN
        END
        INTEGER FUNCTION FIND(TEMP,MISSIONS,NUMISS)
*************************************************************************
*     FUNCTION:  SEARCH FOR OCCURANCE OF MISSION NAME IN DATABASE       *
*                                                                       *
*     AUTHOR    :  LT J.E. TOMKO, USCG                                  *
```

```
*                                                                     *
*     WRITTEN   :  18 MAY 1991                                        *
*                                                                     *
*     MODIFIED  :  18 MAY 1991                                        *
***********************************************************************
* PURPOSE:  CONDUCT A SEQUENTIAL SEARCH OF SMALL DATABASE (ONLY 10    *
* ENTRIES) TO FIND IF A MISSION NAME, CHARACTER STRING OF LENGTH 5,   *
* IS LOCATED WITHIN THE DATABASE.  PASS BACK A VALUE OF 0 IF NOT      *
* FOUND, OTHERWISE PASS BACK THE LOCATION OF THE NAME WITHIN THE      *
* ARRAY.                                                              *
***********************************************************************
      INTEGER NUMISS, I
      CHARACTER TEMP*5, MISSIONS(10)*5
      FIND = 0
      DO 10 I=1,NUMISS
        IF (TEMP .EQ. MISSIONS(I)) FIND = I
10    CONTINUE
      RETURN
      END
      SUBROUTINE ADDMISS(NUMDIST,NUMPLAT,NUMISS,HOURS,MISSIONS,
     +                   DISTRICT,PLATFORM,ATOS,STOA,CATA,VALA,WORDA)
***********************************************************************
*                                                                     *
*     SUBROUTINE:  ADD MISSION DATA FOR NEW MISSION                   *
*                                                                     *
*     AUTHOR    :  LT J.E. TOMKO, USCG                                *
*                                                                     *
*     WRITTEN   :  15 JUNE 1991                                       *
*                                                                     *
*     MODIFIED  :  16 JUNE 1991                                       *
***********************************************************************
* PURPOSE:  COLLECT NEEDED DATA ASSOCIATED WITH AN ADDING A NEW       *
* MISSION.  THIS INCLUDES THE NUMBER OF HOURS EACH DISTRICT REQUIRES  *
* AND THE ABILITY OF THE VARIOUS PLATFORMS TO PERFORM THIS MISSION.   *
***********************************************************************
      INTEGER I, J, K, NUMPLAT(4), NUMDIST
      INTEGER NUMISS, HOURS(7,10,6)
      INTEGER CATA(24,10)
      REAL THOUR(3), ATOS(10), STOA(10), PCT, VALA(4), TCAT
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, STR(2)*7
      CHARACTER PLATFORM(24)*8, ANS*1, WORDA(4)*18
      LOGICAL ERROR
      DATA STR/'SURFACE','AIR    '/
      WRITE(*,11) MISSIONS(NUMISS)
      DO 10 I = 1,NUMDIST
        DO 20 K = 1,2
          ERROR = .TRUE.
          DO 30 WHILE (ERROR)
            WRITE(*,22) DISTRICT(I), STR(K)
            READ(*,*) (THOUR(J), J=1,3)
            IF ((THOUR(1) .LT. 0.0) .OR. (THOUR(2) .LT. 0.0) .OR.
     +          (THOUR(3) .LT. 0.0)) THEN
```

```
              WRITE(*,33)
            ELSE
              DO 40 J = 1,3
                HOURS(I,NUMISS,J+(K-1)*3) = INT(THOUR(J))
40            CONTINUE
              ERROR = .FALSE.
            ENDIF
30        ENDDO
20      CONTINUE
10    CONTINUE
      DO 50 K = 1,2
        ERROR = .TRUE.
        DO 60 WHILE (ERROR)
          WRITE(*,44) STR(K), STR(3-K)
          READ(*,*) PCT
          IF ((PCT .LT. 0.0) .OR. (PCT .GT. 1.0)) THEN
            WRITE(*,55)
          ELSE
            ERROR = .FALSE.
            IF (K .EQ. 1) THEN
              ATOS(NUMISS) = PCT
            ELSE
              STOA(NUMISS) = PCT
            ENDIF
          ENDIF
60      ENDDO
50    CONTINUE
      DO 70 I = 1, NUMPLAT(1)
        ERROR = .TRUE.
        DO 80 WHILE (ERROR)
          WRITE(*,66) PLATFORM(I), MISSIONS(NUMISS),
     +                (J, WORDA(J), VALA(J), J=1,4)
          READ(*,*) TCAT
          IF ((TCAT .LT. 1.0) .OR. (TCAT .GT. 4)) THEN
            WRITE(*,55)
          ELSE
            CATA(I,NUMISS) = INT(TCAT)
            ERROR = .FALSE.
          ENDIF
80      ENDDO
70    CONTINUE
11    FORMAT(' ','FOR MISSION ( ',A5,' ) ENTER THE FOLLOWING DATA:',/
     +       /,' MISSION RESOURCE HOUR REQUIREMENTS (SURFACE & AIR):')
22    FORMAT(/,' FOR DISTRICT ( ',A10,' ) ENTER THE NUMBER OF ',A7,
     +       /,' RESOURCE HOURS NEEDED FOR EACH CLASS (1,2,3):',
     +       /,' ENTER 3 WHOLE NUMBERS SEPERATED BY COMMAS:')
33    FORMAT(' ','*** ERROR ***',/,' NEGATIVE VALUES NOT ALLOWED.',/)
44    FORMAT(' ','ENTER THE PERCENTAGE OF ',A7,' RESOURCE HOURS ',
     +       'REQUIRED',/,' WHICH CAN BE DONE BY ',A7,' PLATFORMS:',/,
     +       ' (ENTER A NUMBER BETWEEN 0.0 AND 1.0):')
55    FORMAT(' ','*** ERROR ***',/,' INVALID ENTRY.',/)
66    FORMAT(' ','ENTER THE RATING FOR PLATFORM TYPE: ',A8,/
```

```
     +       ' CONCERNING MISSION AREA: ',A5,/,' THE FOLLOWING ',
     +       'RATINGS ARE AVAILABLE:',/,' RATING',2X,'DESCRIPTION',11X,
     +       'VALUE',/,1X,6('-'),2X,17('-'),5X,5('-'),4(/,3X,I1,5X,
     +        A18,4X,F4.2),/,' ENTER RATING (1-4):')
      RETURN
      END
      SUBROUTINE CHGMISS(NUMDIST,NUMPLAT,NUMISS,HOURS,MISSIONS,
     +                   DISTRICT,PLATFORM,ATOS,STOA,CATA,VALA,
     +                   WORDA,LOCATE,TEMP)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  CHANGE MISSION NAME OR DATA FOR CURRENT MISSION      *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  16 JUNE 1991                                        *
*                                                                       *
*     MODIFIED   :  01 SEPTEMBER 1991                                   *
*************************************************************************
* PURPOSE:  ALLOW MODIFCATION TO DATA ASSOCIATED WITH A CURRENT         *
* MISSION.  THIS INCLUDES THE NUMBER OF HOURS EACH DISTRICT REQUIRES    *
* AND THE ABILITY OF THE VARIOUS PLATFORMS TO PERFORM THIS MISSION.     *
*************************************************************************
      INTEGER I, J, K, NUMPLAT(4), NUMDIST, LOCATE, FIND, START, END
      INTEGER NUMISS, HOURS(7,10,6), CATA(24,10), HOLD, PICK
      REAL THOUR(3), ATOS(10), STOA(10), PCT, VALA(4)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, STR(2)*7, OPT*1
      CHARACTER PLATFORM(24)*8, ANS*1, WORDA(4)*18, TEMP*5, TEMP1*5
      LOGICAL ERROR
      DATA STR/'SURFACE','AIR    '/
      ANS = 'A'
      DO 10 WHILE (ANS .NE. 'E')
        WRITE(*,11) MISSIONS(LOCATE)
        READ(*,55) ANS
        IF (ANS .EQ. 'N') THEN
           WRITE(*,22) TEMP
           READ(*,44) TEMP1
           HOLD = FIND(TEMP1,MISSIONS,NUMISS)
           IF (HOLD .EQ. 0) THEN
             MISSIONS(LOCATE) = TEMP1
             TEMP = TEMP1
           ELSE
             WRITE(*,33) TEMP1
           ENDIF
        ELSEIF (ANS .EQ. 'D') THEN
          OPT = 'A'
          DO 20 WHILE (OPT .NE. 'E')
            WRITE(*,77)
            READ(*,55) OPT
            IF (OPT .EQ. 'P') THEN
              PICK = 0
              DO 90 WHILE (PICK .NE. 1)
```

```
              PICK = 0
              DO 100 WHILE ((PICK .LT. 1) .OR.
     +                      (PICK .GT. NUMPLAT(1)+2))
                WRITE(*,155) (I+2, PLATFORM(I), I=1,NUMPLAT(1))
                WRITE(*,166) NUMPLAT(1)+2
                READ(*,*) PICK
                IF ((PICK .LT. 1) .OR. (PICK .GT. NUMPLAT(1)+2)) THEN
                  WRITE(*,111)
                ENDIF
100           ENDDO
              IF (PICK .GT. 1) THEN
                IF (PICK .EQ. 2) THEN
                  START = 1
                  END = NUMPLAT(1)
                ELSE
                  START = PICK-2
                  END = PICK-2
                ENDIF
                DO 70 I = START, END
                  ERROR = .TRUE.
                  DO 80 WHILE (ERROR)
                    WRITE(*,122) PLATFORM(I), MISSIONS(LOCATE),
     +                           WORDA(CATA(I,LOCATE)),
     +                           (J, WORDA(J), VALA(J), J=1,4)
                    READ(*,*) TCAT
                    IF ((TCAT .LT. 1.0) .OR. (TCAT .GT. 4)) THEN
                      WRITE(*,111)
                    ELSE
                      CATA(I,LOCATE) = INT(TCAT)
                      ERROR = .FALSE.
                    ENDIF
80                ENDDO
70              CONTINUE
                ENDIF
90            ENDDO
            ELSEIF (OPT .EQ. 'D') THEN
              PICK = 0
              DO 190 WHILE (PICK .NE. 1)
                PICK = 0
                DO 150 WHILE ((PICK .LT. 1) .OR.
     +                       (PICK .GT. NUMDIST+2))
                  WRITE(*,199) (I+2, DISTRICT(I), I=1,NUMDIST)
                  WRITE(*,166) NUMDIST+2
                  READ(*,*) PICK
                  IF ((PICK .LT. 1) .OR. (PICK .GT. NUMDIST+2)) THEN
                    WRITE(*,111)
                  ENDIF
150             ENDDO
                IF (PICK .GT. 1) THEN
                  IF (PICK .EQ. 2) THEN
                    START = 1
                    END = NUMDIST
```

```
                ELSE
                  START = PICK-2
                  END = PICK-2
                ENDIF
                DO 110 I = START, END
                  DO 120 K = 1,2
                    ERROR = .TRUE.
                    DO 130 WHILE (ERROR)
                      WRITE(*,211) DISTRICT(I), STR(K),
     +                        (HOURS(I,LOCATE,J+(K-1)*3), J=1,3),
     +                        TEMP
                      READ(*,*) (THOUR(J), J=1,3)
                      IF ((THOUR(1) .LT. 0.0) .OR.
     +                    (THOUR(2) .LT. 0.0) .OR.
     +                    (THOUR(3) .LT. 0.0)) THEN
                        WRITE(*,177)
                      ELSE
                        DO 140 J = 1,3
                          HOURS(I,LOCATE,J+(K-1)*3) = INT(THOUR(J))
140                     CONTINUE
                        ERROR = .FALSE.
                      ENDIF
130                 ENDDO
120                 CONTINUE
110               CONTINUE
                ENDIF
190           ENDDO
            ELSEIF (OPT .EQ. 'T') THEN
              DO 50 K = 1,2
                ERROR = .TRUE.
                DO 60 WHILE (ERROR)
                  WRITE(*,88) STR(K), STR(3-K)
                  IF (K .EQ. 1) WRITE(*,99) ATOS(LOCATE)
                  IF (K .EQ. 2) WRITE(*,99) STOA(LOCATE)
                  READ(*,*) PCT
                  IF ((PCT .LT. 0.0) .OR. (PCT .GT. 1.0)) THEN
                    WRITE(*,111)
                  ELSE
                    ERROR = .FALSE.
                    IF (K .EQ. 1) THEN
                      ATOS(LOCATE) = PCT
                    ELSE
                      STOA(LOCATE) = PCT
                    ENDIF
                  ENDIF
60              ENDDO
50            CONTINUE
            ELSEIF (OPT .NE. 'E') THEN
              WRITE(*,66) OPT
            ENDIF
20        ENDDO
        ELSEIF (ANS .NE. 'E') THEN
```

```
          WRITE(*,66) ANS
        ENDIF
10     ENDDO
11     FORMAT(' ','FOR MISSION ( ',A5,' ) YOU CAN CHANGE EITHER:',/,
     +       ' (N)AME OF MISSION',/,' (D)ATA CONCERNING THE MISSION',/,
     +       ' (E)XIT THIS ROUTINE',/,' ENTER YOUR CHOICE (N,D,E):')
22     FORMAT(' ','ENTER NEW NAME FOR MISSION (',A5,').')
33     FORMAT(' ','*** ERROR ***',/,' MISSION NAME: ',A5,' ALREADY',
     +       ' EXISTS.')
44     FORMAT(A5)
55     FORMAT(A1)
66     FORMAT(' ','*** ERROR ***',/,' (',A1,') IS NOT A VALID ENTRY',/)
77     FORMAT(' ','YOU CAN CHANGE THE FOLLOWING MISSION DATA:',/,
     +       ' (P)LATFORM RATING',/,' (D)ISTRICT REQUIRED HOURS',/,
     +       ' (T)RADE-OFFS BETWEEN AIR & SURFACE PLATFORMS',/,' (E)',
     +       'XIT ROUTINE',/,' ENTER YOUR CHOICE (P,D,T,E):')
88     FORMAT(' ','ENTER THE PERCENTAGE OF ',A7,' RESOURCE HOURS ',
     +       'REQUIRED',/,' WHICH CAN BE DONE BY ',A7,' PLATFORMS.')
99     FORMAT(' ','CURRENT PERCENTAGE IS : ',F4.2,/,' ENTER NEW ',
     +       'PERCENTAGE (ANY NUMBER BETWEEN 0.0 AND 1.0):')
111    FORMAT(' ','*** ERROR ***',/,' INVALID ENTRY.',/)
122    FORMAT(' ','THE CURRENT RATING FOR PLATFORM TYPE: ',A8,/
     +       ' CONCERNING MISSION AREA: ',A5,' IS ',A18,/,' THE ',
     +       'FOLLOWING RATINGS ARE AVAILABLE:',/,' RATING',2X,
     +       'DESCRIPTION',11X,'VALUE',/,1X,6('-'),2X,17('-'),5X,
     +       5('-'),4(/,3X,I1,5X,A18,4X,F4.2),/,
     +       ' ENTER NEW RATING (1-4):')
155    FORMAT(' ','YOU CAN CHANGE THE RATING OF THE FOLLOWING ',
     +       'PLATFORMS:',/,'  1) EXIT      2) ALL',5X,3(1X,I2,') ',A8)
     +       ,/,5(5(1X,I2,') ',A8),/))
166    FORMAT(' ','ENTER YOUR CHOICE FOR MODIFICATION (1 - ',I2,'):')
177    FORMAT(' ','*** ERROR ***',/,' NEGATIVE VALUES NOT ALLOWED',/)
188    FORMAT(/,' FOR DISTRICT ( ',A10,' ) ENTER THE NUMBER OF ',A7,
     +       /,' RESOURCE HOURS NEEDED AT EACH CLASS (1,2,3):',
     +       /,' ENTER 3 WHOLE NUMBERS SEPERATED BY COMMAS:')
199    FORMAT(' ','YOU CAN CHANGE THE MISSION REQUIREMENTS FOR THE',/,
     +       ' FOLLOWING PLATFORMS:',/,'  1) EXIT',8X,'2) ALL',7X,
     +       3(1X,I2,') ',A10),/,4(1X,I2,') ',A10))
211    FORMAT(/,' DISTRICT ( ',A10,' ) CURRENTLY REQUIRES THE FOLLOWING'
     +       ,1X,A7,/,' RESOURCE HOURS FOR EACH CLASS (1,2,3):',
     +       /,6X,'1',10X,'2',10X,'3',/,3(2X,9('-')),/,3(3X,I5,3X),
     +       /,' ENTER NEW RESOURCE HOURS REQUIRED FOR MISSION: ',
     +       A5,/,' (3 WHOLE NUMBERS SEPERATED BY COMMAS):')
       RETURN
       END
       SUBROUTINE CHANGE(WORDA,WORDB,VALA,VALB,ATOS,STOA,NUMPLAT,
     +                   NUMISS,MISSIONS,PLATFORM)
*****************************************************************************
*                                                                           *
*     SUBROUTINE:  CHANGE PARAMETER DATA                                    *
*                                                                           *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                     *
```

```
*                                                                     *
*     WRITTEN   :  07 JUNE 1991                                       *
*                                                                     *
*     MODIFIED  :  09 JUNE 1991                                       *
***********************************************************************
* PURPOSE:  MENU DRIVEN USER INTERFACE TO ALLOW FOR MODIFICATION OF   *
* THE TABULAR DATA NEEDED BY GAMS.  THIS INCLUDES THE PERCENTAGE OF   *
* RESOURCE HOURS THAT CAN BE PERFORMED BY DIFFERENT PLATFORM TYPES    *
* (IE ATOS & STOA), THE VALUES AND DESCRIPTORS FOR THE MISSION        *
* PERFORMANCE LEVELS (VALB & WORDB) AND THE THREE CLASSES (VALA &     *
* WORDA)                                                              *
***********************************************************************
      INTEGER NUMISS, NUMPLAT(4), I
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), PCT
      CHARACTER MISSIONS(10)*5, PLATFORM(24)*8, ANS*1, OPT*1
      CHARACTER WORDA(4)*18, WORDB(4)*18
      LOGICAL ERROR
      ANS = 'A'
      DO 10 WHILE (ANS .NE. 'E')
        ERROR = .TRUE.
        WRITE(*,11)
        READ(*,22) ANS
        IF (ANS .EQ. 'A') THEN
          CALL CTRADE(ATOS,NUMISS,MISSIONS,ANS)
        ELSEIF (ANS .EQ. 'S') THEN
          CALL CTRADE(STOA,NUMISS,MISSIONS,ANS)
        ELSEIF (ANS .EQ. 'C') THEN
          CALL CVAL(VALB,WORDB,ANS)
        ELSEIF (ANS .EQ. 'M') THEN
          CALL CVAL(VALA,WORDA,ANS)
        ELSEIF (ANS .EQ. 'E') THEN
          PRINT*,'EXITING CHANGE PARAMETERS ROUTINE'
        ELSE
          PRINT*,'*** ERROR ***'
          PRINT*,'PLEASE ENTER A,S,C,M,P OR E'
        ENDIF
10    ENDDO
11    FORMAT('/',' THE FOLLOWING DATA MAY BE MODIFIED:'/,' (A)IR TO',
     +       ' SURFACE TRADE-OFFS',/,' (S)URFACE TO AIR TRADE-OFFS',/,
     +       ' (C)APABILITY PERCENTAGES',/,' (M)ISSION PERCENTAGES',/,
     +       ' (E)XIT EDIT ROUTINE',/,' ENTER YOUR CHOICE (A,S,C,M,E):')
22    FORMAT(A1)
      RETURN
      END
      SUBROUTINE CTRADE(PCNT,NUMISS,MISSIONS,ANS)
***********************************************************************
*                                                                     *
*     SUBROUTINE:  CHANGE TRADE-OFF DATA                              *
*                                                                     *
*     AUTHOR    :  LT J.E. TOMKO, USCG                                *
*                                                                     *
*     WRITTEN   :  07 JUNE 1991                                       *
```

```
*                                                                      *
*     MODIFIED  :  09 JUNE 1991                                        *
************************************************************************
* PURPOSE:  MENU DRIVEN USER INTERFACE TO ALLOW FOR MODIFICATION OF    *
* THE PERCENTAGE OF RESOURCE HOURS THAT CAN BE PERFORMED BY            *
* DIFFERENT PLATFORM TYPES (IE ATOS & STOA)                            *
************************************************************************
      INTEGER NUMISS, I, PICK, START, END
      REAL PCNT(10), PCT
      CHARACTER MISSIONS(10)*5, ANS*1, OPT*1
      LOGICAL ERROR
      OPT = 'A'
      DO 10 WHILE (OPT .NE. 'E')
        IF (ANS .EQ. 'A') THEN
          WRITE(*,11) (MISSIONS(I), I=1,NUMISS)
        ELSE
          WRITE(*,44) (MISSIONS(I), I=1,NUMISS)
        ENDIF
        WRITE(*,33) (PCNT(I), I=1,NUMISS)
        WRITE(*,55)
        READ(*,22) OPT
        IF (OPT .EQ. 'A') THEN
          ERROR = .TRUE.
          DO 20 WHILE (ERROR)
            WRITE(*,66)
            READ(*,*) PCT
            IF ((PCT .LT. 0) .OR. (PCT .GT. 1)) THEN
              WRITE(*,77)
            ELSE
              ERROR = .FALSE.
              DO 30 I = 1, NUMISS
                PCNT(I) = PCT
30            CONTINUE
              IF (ANS .EQ. 'A') THEN
                WRITE(*,11) (MISSIONS(I), I=1,NUMISS)
              ELSE
                WRITE(*,44) (MISSIONS(I), I=1,NUMISS)
              ENDIF
              WRITE(*,33) (PCNT(I), I=1,NUMISS)
            ENDIF
20        ENDDO
        ELSEIF (OPT .EQ. 'B') THEN
          ERROR = .TRUE.
          DO 50 WHILE (ERROR)
            WRITE(*,88) (I+1,MISSIONS(I), I=1,NUMISS)
            WRITE(*,99) NUMISS+1
            READ(*,*) PICK
            IF ((PICK .LT. 1) .OR. (PICK .GT. NUMISS+1)) THEN
              WRITE(*,77)
            ELSE
              ERROR = .FALSE.
            ENDIF
```

```
50          ENDDO
            IF (PICK .EQ. 1) THEN
              START = 1
              END = NUMISS
            ELSE
              START = PICK-1
              END = PICK-1
            ENDIF
            DO 40 I = START, END
              ERROR = .TRUE.
              DO 80 WHILE (ERROR)
                IF (ANS .EQ. 'A') THEN
                  WRITE(*,111) PCNT(I), MISSIONS(I)
                ELSE
                  WRITE(*,122) PCNT(I), MISSIONS(I)
                ENDIF
                WRITE(*,66)
                READ(*,*) PCT
                IF ((PCT .LT. 0) .OR. (PCT .GT. 1)) THEN
                  WRITE(*,77)
                ELSE
                  ERROR = .FALSE.
                  PCNT(I) = PCT
                ENDIF
80            ENDDO
40          CONTINUE
          ENDIF
10      ENDDO
11      FORMAT(' ','AIRCRAFT CAN CURRENTLY DO THE FOLLOWING PERCENTAGE'
     +         ,/,' OF SURFACE RESOURCE HOURS IN EACH MISSION:'
     +         ,/,6X,10(A5,2X))
22      FORMAT(A1)
33      FORMAT(5X,10(F5.2,2X),/)
44      FORMAT(' ','CUTTERS CAN CURRENTLY DO THE FOLLOWING PERCENTAGE'
     +         ,/,' OF AIR RESOURCE HOURS IN EACH MISSION:'
     +         ,/,6X,10(A5,2X))
55      FORMAT('/',' YOU MAY EITHER:',/,' (A) CHANGE ALL PERCENTAGES ',
     +         'TO A SINGLE VALUE',/,' (B) CHANGE EACH MISSION VALUE ',
     +         'INDIVIDUALLY',/,' (E)XIT THIS SUBROUTINE',/,
     +         ' PLEASE ENTER CHOICE A, B OR E:')
66      FORMAT(' ','ENTER NEW PERCENTAGE (BETWEEN 0.0 AND 1.0):')
77      FORMAT(' ','*** ERROR ***',/,' INVALID ENTRY.',/)
88      FORMAT(' ','YOU MAY MODIFY THE FOLLOWING MISSION AREAS:',/,
     +         5X,' 1) ALL  ',5(I2,') ',A5),/,5X,5(I2,') ',A5))
99      FORMAT(' ','ENTER YOUR CHOICE FOR MODIFICATION (1 -',I2,'):')
111     FORMAT(' ','AIRCRAFT CAN CURRENTLY DO A MAXIMUM OF ',F6.2,' OF',
     +         ' SURFACE',/,' RESOURCE HOUR REQUIREMENTS IN THE MISSION',
     +         ' AREA ( ',A5,' ).')
122     FORMAT(' ','CUTTERS CAN CURRENTLY DO A MAXIMUM OF ',F6.2,' OF',
     +         ' AIR RESOURCE',/,' HOUR REQUIREMENTS IN THE MISSION',
     +         ' AREA ( ',A5,' ).')
        RETURN
```

```
      END
      SUBROUTINE CVAL(VAL,WORD,ANS)
*************************************************************************
*                                                                       *
*     SUBROUTINE:  CHANGE ASSIGNMENT VALUES AND DESCRIPTORS             *
*                                                                       *
*     AUTHOR     :  LT J.E. TOMKO, USCG                                 *
*                                                                       *
*     WRITTEN    :  09 JUNE 1991                                        *
*                                                                       *
*     MODIFIED   :  09 JUNE 1991                                        *
*************************************************************************
* PURPOSE:  MENU DRIVEN USER INTERFACE TO ALLOW FOR MODIFICATION OF     *
* THE VALUES ASSIGNED TO  THE 4 POSSIBLE  MISSION ASSIGNMENT LEVELS     *
* AND THE 4 POSSIBLE CLASS PERFORMANCE GRADES.                          *
* THE DESCRIPTORS FOR THESE LEVELS MAY BE CHANGED BY THE USERS.         *
*************************************************************************
      INTEGER I, PICK
      REAL VAL(4), NUM, NVAL
      CHARACTER WORD(4)*18, ANS*1, OPT*1, NWORD*18
      LOGICAL ERROR
      OPT = 'A'
      DO 10 WHILE (OPT .NE. 'E')
        IF (ANS .EQ. 'M') THEN
          WRITE(*,11)
        ELSE
          WRITE(*,22)
        ENDIF
        WRITE(*,33)
        DO 20 I = 1,4
          WRITE(*,44)  I, VAL(I), WORD(I)
20      CONTINUE
        WRITE(*,55)
        READ(*,66) OPT
        IF (OPT .EQ. 'C') THEN
          ERROR = .TRUE.
          DO 30 WHILE (ERROR)
            WRITE(*,77)
            READ(*,*) NUM
            PICK = INT(NUM)
            IF (( PICK .LT. 1) .OR. (PICK .GT. 4)) THEN
              WRITE(*,88) NUM
            ELSE
              WRITE(*,99) PICK, VAL(PICK), WORD(PICK)
              READ(*,66) OPT
              IF (OPT .EQ. 'V') THEN
                DO 40 WHILE (ERROR)
                  WRITE(*,122) PICK, VAL(PICK)
                  READ(*,*) NVAL
                  IF ((NVAL .LT. 0) .OR. (NVAL .GT. 1)) THEN
                    WRITE(*,88) NVAL
                  ELSE
```

```
                    ERROR = .FALSE.
                    VAL(PICK) = NVAL
                  ENDIF
40               ENDDO
               ELSEIF (OPT .EQ. 'D') THEN
                 DO 50 WHILE (ERROR)
                   ERROR = .FALSE.
                   WRITE(*,133) PICK, WORD(PICK)
                   READ(*,144) NWORD
                   DO 60 I = 1,4
                     IF (NWORD .EQ. WORD(I)) ERROR = .TRUE.
60                 CONTINUE
                   IF (ERROR) THEN
                     WRITE(*,155) NWORD
                   ELSE
                     WORD(PICK) = NWORD
                   ENDIF
50               ENDDO
               ELSE
                 WRITE(*,111) OPT
               ENDIF
             ENDIF
30         ENDDO
         ELSEIF (OPT .NE. 'E') THEN
           WRITE(*,111) OPT
         ENDIF
10     ENDDO
11     FORMAT(' ','THE FOLLOWING VALUES AND DESCRIPTIONS ARE DEFINED ',
     +       'FOR',/,' MISSION ASSIGNMENT CODES:')
22     FORMAT(' ','THE FOLLOWING VALUES AND DESCRIPTIONS ARE DEFINED ',
     +       'FOR',/,' CLASS CODES:')
33     FORMAT(' ',5X,'CODE',3X,'VALUE',8X,'DESCRIPTION',/,6X,4('-'),
     +       3X,5('-'),4X,18('-'))
44     FORMAT(' ',7X,I1,5X,F4.2,5X,A18)
55     FORMAT(/,' THE FOLLOWING OPTIONS ARE AVAILABLE:',/,' (C)HANGE ',
     +       'THE VALUE OR DESCRIPTION FOR A CODE',/,' (E)XIT THIS ',
     +       'ROUTINE',/,' PLEASE ENTER YOUR CHOICE (C OR E):')
66     FORMAT(A1)
77     FORMAT(' ','ENTER THE CODE NUMBER YOU WISH TO EDIT (1-4):')
88     FORMAT(' ','*** ERROR ***',/,' ( ',F7.2,'  ) IS NOT VALID')
99     FORMAT(/,' FOR CODE (',I1,'):',/,' THE VALUE IS (',F4.2,') ',
     +         'WITH A DESCRIPTION OF (',A18,')',//,' YOU MAY EDIT',
     +         ' EITHER THE:',/,' (V)ALUE',/,' (D)ESCRIPTION',/,
     +         ' PLEASE ENTER YOUR CHOICE (V OR D):')
111    FORMAT(' ','*** ERROR ***',/,' (',A1,') IS NOT A VALID ENTRY.',/)
122    FORMAT(' ','CURRENT VALUE FOR CODE (',I1,') IS ',F4.2,/,
     +          ' ENTER NEW VALUE BETWEEN 0.0 AND 1.0:')
133    FORMAT(' ','CURRENT DESCRIPTION FOR CODE (',I1,') IS ',A18,/,
     +          ' ENTER NEW DESCRIPTION WITH NO MORE ',
     +          'THAN 18 CHARACTERS:')
144    FORMAT(A18)
155    FORMAT(' ','*** ERROR ***',/,' DESCRIPTION (',A18,') ALREADY ',
```

```
     +       'EXISTS.',/)
      RETURN
      END
   SUBROUTINE DOIT(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,
  +                MISSIONS,DISTRICT,PLATFORM,TYPE,CATA,CATB,
  +                VALA,VALB,ATOS,STOA,QTY,PCNT,INDP,LOCINDP,MODEL)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  PRODUCE INCLUDE FILE FOR GAMS APPLICATION              *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  03 JUNE 1991                                          *
*                                                                       *
*   MODIFIED   :  30 AUGUST 1991                                        *
*                                                                       *
*************************************************************************
*  PURPOSE:  PRODUCE THE VARIOUS ELEMENTS OF THE GAMS INCLUDE FILE      *
*  FOR RUNNING THE GAMS PROGRAM.  THE INDIVIDUAL ELEMENTS ARE           *
*  PRODUCED IN SEPARATE SUBROUTINES CONTROLLED BY THIS SUBROUTINE.      *
*  SINCE THE ACTUAL GAMS MODELS ARE DIFFERENT AND THE ORIGINAL DATA     *
*  USED BY THE MODELS IS THE SAME ONLY ONE DATA FILE WILL BE            *
*  PRODUCED.                                                            *
*************************************************************************
      INTEGER  NUMDIST, NUMPLAT(4), NUMISS, HRS(24), INDP, LOCINDP(7)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      REAL COST(24), MODEL(7)
      REAL VALA(4), VALB(4), ATOS(10), STOA(10)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER TYPE(24)*3, ANS*1, CG*26, TEMP*8
      CHARACTER CODE(3)*3, STR(3)*16
      DATA STR/'DISTRICT SURFACE','DISTRICT AIR    ','AREA SURFACE    '/
      DATA CODE/'IDS', 'IDA', 'IAS'/
      OPEN(30,FILE='/GAMS DATA A1')
      CALL INFEAS(NUMPLAT,HOURS,NUMDIST,NUMISS,HRS,VALA,VALB,
     +            CATA,CATB,MISSIONS)
      CALL SETS(NUMPLAT,NUMDIST,NUMISS,PLATFORM,DISTRICT,MISSIONS,
     +          TYPE,CODE,STR,INDP,LOCINDP)
      CALL TABLES(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,
     +            MISSIONS,DISTRICT,PLATFORM,TYPE,CATA,CATB,VALA,
     +            VALB,ATOS,STOA,QTY,CODE)
      CALL SCALARS
      CALL PARAMETERS(NUMPLAT,NUMISS,HRS,HOURS,COST,MISSIONS,
     +                PLATFORM,TYPE,CATA,CATB,VALA,VALB,ATOS,
     +                STOA,QTY,CODE)
      CALL EQUATIONS(NUMPLAT,MODEL)
      CLOSE(30)
      RETURN
      END
```

```
      SUBROUTINE SETS(NUMPLAT,NUMDIST,NUMISS,PLATFORM,DISTRICT,
     +                MISSIONS,TYPE,CODE,STR,INDP,LOCINDP)
************************************************************************
*                                                                      *
*   SUBROUTINE:  WRITE SET DATA TO INCLUDE FILE FOR GAMS MODEL         *
*                                                                      *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                  *
*                                                                      *
*   WRITTEN    :  03 JUNE 1991                                         *
*                                                                      *
*   MODIFIED   :  09 JUNE 1991                                         *
*                                                                      *
************************************************************************
*  PURPOSE: WRITE THE SET DATA TO THE INCLUDE FILE FOR THE GAMS        *
*  PROGRAM.                                                            *
************************************************************************
      INTEGER I, J, K, LOOP, NUMDIST, NUMPLAT(4), NUMISS, CNT
      INTEGER INDP, LOCINDP(7), HOLD(3)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER PLATS(24)*9, A*1, B*1, MISS(10)*6, DIST(7)*11
      CHARACTER CODE(3)*3, TYPE(24)*3, STR(3)*16, BLANK(3)*5
      DATA BLANK/'BLANK', 'BLANK', 'BLANK'/
      A = ','
      B = '/'
      DO 5 I = 1,NUMPLAT(1) - 1
        PLATS(I) = PLATFORM(I)//A
5     CONTINUE
      PLATS(NUMPLAT(1)) = PLATFORM(NUMPLAT(1))//B
      WRITE(30,11)
      LOOP = NUMPLAT(1)/6
      CNT = 0
      DO 40 I = 2,4
        IF (NUMPLAT(I) .EQ. 0) THEN
          CNT = CNT + 1
          HOLD(CNT) = I
        ENDIF
40    CONTINUE
      IF (CNT .GT. 0) THEN
        WRITE(30,122) (BLANK(I),HOLD(I), I=1,CNT)
      ENDIF
      DO 10 I = 1, LOOP
        WRITE(30,22) (PLATS(J), J=1+(I-1)*6,6+(I-1)*6)
10    CONTINUE
      IF (MOD(NUMPLAT(1),6) .NE. 0) THEN
        WRITE(30,22) (PLATS(J), J=LOOP*6+1,NUMPLAT(1))
      ENDIF
      DO 20 J = 2,4
        WRITE(30,112) CODE(J-1),STR(J-1)
        I = 1
        CNT = 0
        IF (NUMPLAT(J) .EQ. 0) THEN
          WRITE(30,99) J
```

```
        ENDIF
        DO 30 WHILE (CNT .NE. NUMPLAT(J))
          IF (TYPE(I) .EQ. CODE(J-1)) THEN
            CNT = CNT + 1
            IF (CNT .LT. NUMPLAT(J)) THEN
              WRITE(30,55) PLATFORM(I)
            ELSE
              WRITE(30,66) PLATFORM(I)
            ENDIF
          ENDIF
          I = I + 1
30      ENDDO
20    CONTINUE
      DO 60 I = 1,NUMDIST - 1
        DIST(I) = DISTRICT(I)//A
60    CONTINUE
      DIST(NUMDIST) = DISTRICT(NUMDIST)//B
      WRITE(30,115)
      LOOP = NUMDIST/4
      DO 70 I = 1, LOOP
        WRITE(30,44) (DIST(J), J=1+(I-1)*4,4+(I-1)*4)
70    CONTINUE
      IF (MOD(NUMDIST,4) .NE. 0) THEN
        WRITE(30,44) (DIST(J), J=LOOP*4+1,NUMDIST)
      ENDIF
      DO 80 I = 1,NUMISS - 1
        MISS(I) = MISSIONS(I)//A
80    CONTINUE
      MISS(NUMISS) = MISSIONS(NUMISS)//B
      WRITE(30,116)
      WRITE(30,33) (MISS(J), J=1,NUMISS)
      WRITE(30,117)
      IF (INDP .EQ. 0) THEN
        WRITE(30,119)
      ELSE
        WRITE(30,118)
        DO 90 I = 1, INDP
          IF (I .NE. INDP) THEN
            WRITE(30,77) DISTRICT(LOCINDP(I))
          ELSE
            WRITE(30,88) DISTRICT(LOCINDP(I))
          ENDIF
90      CONTINUE
      ENDIF
      WRITE(30,133)
11    FORMAT(' ',' SETS',//,'  P   ALL PLATFORMS',/,'  /')
22    FORMAT(' ',3X,6(A9))
33    FORMAT(2X,'/',1X,10(A6))
44    FORMAT(' ',3X,4(A11))
55    FORMAT(6X,A8,',')
66    FORMAT(6X,A8,2X,'/')
77    FORMAT(43X,A10,3X,',')
```

```
88    FORMAT(43X,A10,3X,'/;',/)
99    FORMAT(6X,'BLANK',I1,2X,'/')
111   FORMAT(/,'  SET INDP(D)',5X,'/')
112   FORMAT(/,3X,A3,'(P)  ',A16,' PLATFORMS',/,'  /')
115   FORMAT(/,'  D  DISTRICT',/,'  /')
116   FORMAT(/,'  M  MISSION')
117   FORMAT(/,'  L  CLASS',10X,'  /1, 2, 3/;',/)
118   FORMAT(/,'  SET INDP(D)  INDEPENDENT DISTRICTS',5X,'/')
119   FORMAT(/,'  SET INDP(D)  INDEPENDENT DISTRICTS;',/
     +        ,6X,'INDP(D) = NO;',/)
122   FORMAT(3X,3(A5,I1,1X,','))
133   FORMAT(/,6X,'ALIAS(D,DP);',/)
      RETURN
      END
      SUBROUTINE TABLES(NUMDIST,NUMPLAT,NUMISS,HRS,HOURS,COST,
     +                  MISSIONS,DISTRICT,PLATFORM,TYPE,CATA,
     +                  CATB,VALA,VALB,ATOS,STOA,QTY,CODE)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  WRITE TABLE DATA TO INCLUDE FILE FOR GAMS MODEL        *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  03 MAY 1991                                           *
*                                                                       *
*   MODIFIED   :  09 JUNE 1991                                          *
*                                                                       *
*************************************************************************
*  PURPOSE: WRITE THE SET DATA TO THE INCLUDE FILE FOR THE GAMS         *
*  PROGRAM.                                                             *
*                                                                       *
*************************************************************************
      INTEGER I, J, K, L, NUMDIST, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24)
      CHARACTER MISSIONS(10)*5, DISTRICT(7)*10, PLATFORM(24)*8
      CHARACTER CODE(3)*3, TYPE(24)*3, TEMP*3
      DO 10 I = 1,3
        TEMP = CODE(I)
        WRITE(30,111) TEMP(2:3), TEMP, TEMP
        WRITE(30,122) (MISSIONS(J), J=1,NUMISS)
        K = 1
        CNT = 0
        IF (NUMPLAT(I+1) .EQ. 0) THEN
          WRITE(30,166) I+1, (0.0, L=1,NUMISS)
        ENDIF
        DO 20 WHILE (CNT .NE. NUMPLAT(I+1))
          IF (TYPE(K) .EQ. CODE(I)) THEN
            CNT = CNT + 1
            WRITE(30,144) PLATFORM(K), (VALA(CATA(K,L)), L=1,NUMISS)
          ENDIF
          K = K + 1
```

```
20      ENDDO
        WRITE(30,99)
        WRITE(30,133) TEMP(2:3), TEMP, TEMP
        K = 1
        CNT = 0
        IF (NUMPLAT(I+1) .EQ. 0) THEN
          WRITE(30,177) I+1, (0.0, L=1,3)
        ENDIF
        DO 30 WHILE (CNT .NE. NUMPLAT(I+1))
          IF (TYPE(K) .EQ. CODE(I)) THEN
            CNT = CNT + 1
            WRITE(30,155) PLATFORM(K), (VALB(CATB(K,L)), L=1,3)
          ENDIF
          K = K + 1
30      ENDDO
        WRITE(30,33)
10    CONTINUE
      WRITE(30,11)
      DO 140 I = 1, NUMDIST
        DO 150 J = 1, NUMISS
          WRITE(30,22) DISTRICT(I),MISSIONS(J),(HOURS(I,J,K),K=1,3)
150     CONTINUE
140   CONTINUE
      WRITE(30,33)
      WRITE(30,44)
      DO 160 I = 1, NUMDIST
        DO 170 J = 1, NUMISS
          WRITE(30,22) DISTRICT(I),MISSIONS(J),(HOURS(I,J,K),K=4,6)
170     CONTINUE
160   CONTINUE
      WRITE(30,33)
      WRITE(30,55)
      IF (NUMDIST .GT. 5) THEN
         WRITE(30,66) (DISTRICT(J), J=1,5)
         DO 180 I = 1, NUMPLAT(1)
           WRITE(30,77) PLATFORM(I), (QTY(I,J), J=1,5)
180      CONTINUE
         WRITE(30,88) (DISTRICT(J), J=6,NUMDIST)
         DO 190 I = 1, NUMPLAT(1)
           WRITE(30,77) PLATFORM(I), (QTY(I,J), J=6,NUMDIST)
190      CONTINUE
      ELSE
         WRITE(30,66) (DISTRICT(J), J=1,NUMDIST)
         DO 200 I = 1, NUMPLAT(1)
           WRITE(30,77) PLATFORM(I), (QTY(I,J), J=1,NUMDIST)
200      CONTINUE
      ENDIF
      WRITE(30,99)
11    FORMAT(3X,'TABLE',/,6X,'HRSURF(D,M,L)  SURFACE RESOURCE HOURS ',
     +     'REQUIRED FOR MISSION (M)',/,'*',19X,'IN DISTRICT (D) AT',
     +     ' CLASS (L) PER YEAR.',/,32X,'1',9X,'2',9X,'3')
22    FORMAT(6X,A10,'.',A5,1X,3(5X,I5))
```

```
33     FORMAT(55X,';')
44     FORMAT(3X,'TABLE',/,6X,'HRAIR(D,M,L)  AIR RESOURCE HOURS ',
      +     'REQUIRED FOR MISSION (M)',/,'*',19X,'IN DISTRICT (D) AT',
      +     ' CLASS (L) PER YEAR.',/,32X,'1',9X,'2',9X,'3')
55     FORMAT(3X,'TABLE',/,6X,'CURRENT(P,D)  CURRENT FORCE STRUCTURE ',
      +       'FOR THE COAST GUARD')
66     FORMAT(13X,5(2X,A10,1X))
77     FORMAT(5X,A8,2X,5(I3,10X))
88     FORMAT(9X,'+',3X,5(2X,A10,1X))
99     FORMAT(71X,';')
111    FORMAT(3X,'TABLE',/,6X,'A',A2,'(',A3,',M)  MAXIMUM PERCENTAGE ',
      +       'OF TIME THAT PLATFORM (',A3,') CAN',/,'*',17X,
      +       'PERFORM MISSION (M).')
122    FORMAT(16X,10(A5,1X))
133    FORMAT(3X,'TABLE',/,6X,'B',A2,'(',A3,',L)  MAXIMUM PERCENTAGE ',
      +       'OF TIME THAT PLATFORM (',A3,') CAN',/,'*',17X,
      +       'PERFORM AT CLASS (L).',/,20X,'1',8X,'2',
      +       8X,'3')
144    FORMAT(6X,A8,2X,10(F4.2,2X))
155    FORMAT(6X,A8,5X,3(F4.2,5X))
166    FORMAT(6X,'BLANK',I1,4X,10(F4.2,2X))
177    FORMAT(6X,'BLANK',I1,7X,3(F4.2,5X))
       RETURN
       END
       SUBROUTINE SCALARS
*************************************************************************
*                                                                       *
*   SUBROUTINE:  WRITE SCALAR DATA TO INCLUDE FILE FOR GAMS MODEL       *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  03 JUNE 1991                                          *
*                                                                       *
*   MODIFIED   :  30 AUGUST 1991                                        *
*                                                                       *
*************************************************************************
*  PURPOSE: WRITE THE SCALAR DEFINITION DATA TO THE INCLUDE FILE        *
*  FOR THE GAMS PROGRAM.                                                *
*                                                                       *
*************************************************************************
       WRITE(30,11)
       WRITE(30,22)
11     FORMAT(' ','SCALARS',/,3X,'PICK',6X,'FLAG FOR WHICH ',
      +       'OBJECTIVE FUNCTION TO USE',/,3X,'CNT',7X,
      +       'PLATFORM COUNT',/,3X,'ACQLIM',4X,'FLAG FOR LIMITING',
      +       ' ACQUISITION COSTS',/,3X,'MOVLIM',4X,'FLAG FOR LIMITING',
      +       ' MOVEMENT COSTS',/,3X,'LIMACQ',4X,'LIMIT FOR ',
      +       'ACQUISITION COSTS',/,3X,'LIMMOV',4X,'LIMIT FOR ',
      +       'MOVEMENT COSTS',/,3X,'PENALTY1',1X,'ACQUISITION COST ',
      +       'CONSTANT',/,3X,'PENALTY2',1X,'MOVEMENT COST CONSTANT',/)
22     FORMAT(' ',2X,'NOIDS     FLAG FOR NO DISTRICT SURFACE PLATFORMS',
      +       ' DEFINED',/,3X,'NOIAS     FLAG FOR NO AREA SURFACE',
```

```
     +       ' PLATFORMS DEFINED',/,3X,'NOIDA     FLAG FOR NO ',
     +       'DISTRICT AIR PLATFORMS DEFINED',/)
      RETURN
      END
      SUBROUTINE PARAMETERS(NUMPLAT,NUMISS,HRS,HOURS,COST,MISSIONS,
     +                      PLATFORM,TYPE,CATA,CATB,VALA,VALB,ATOS,STOA,
     +                      QTY,CODE)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  WRITE PARAMETER DATA TO FILE FOR GAMS MODEL            *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  03 JUNE 1991                                          *
*                                                                       *
*   MODIFIED   :  30 AUGUST 1991                                        *
*                                                                       *
*************************************************************************
*  PURPOSE: WRITE THE PARAMETER DATA TO THE INCLUDE FILE FOR THE        *
*  GAMS PROGRAM.                                                        *
*                                                                       *
*************************************************************************
      INTEGER I, J, K, NUMPLAT(4), NUMISS, HRS(24)
      INTEGER HOURS(7,10,6), QTY(24,7), CATB(24,3), CATA(24,10)
      INTEGER LOCATE, HUNT, CNT
      REAL VALA(4), VALB(4), ATOS(10), STOA(10), COST(24)
      CHARACTER MISSIONS(10)*5, PLATFORM(24)*8
      CHARACTER CODE(3)*3, TYPE(24)*3, TEMP*3
      WRITE(30,11)
      WRITE(30,22)
      WRITE(30,33) MISSIONS(1), ATOS(1)
      DO 10 I = 2, NUMISS-1
        WRITE(30,44) MISSIONS(I), ATOS(I)
10    CONTINUE
      WRITE(30,55) MISSIONS(NUMISS), ATOS(NUMISS)
      WRITE(30,66)
      WRITE(30,33) MISSIONS(1), STOA(1)
      DO 20 I = 2, NUMISS-1
        WRITE(30,44) MISSIONS(I), STOA(I)
20    CONTINUE
      WRITE(30,55) MISSIONS(NUMISS), STOA(NUMISS)
      DO 30 I = 1,3
        TEMP = CODE(I)
        WRITE(30,77) TEMP(2:3), TEMP, TEMP
        IF (NUMPLAT(I+1) .EQ. 1) THEN
          LOCATE = HUNT(TYPE,TEMP)
          WRITE(30,111) PLATFORM(LOCATE), HRS(LOCATE)
        ELSEIF (NUMPLAT(I+1) .EQ. 0) THEN
          WRITE(30,199) I+1, 0
        ELSE
          K = 1
          CNT = 0
```

```
          DO 50 WHILE (CNT .NE. NUMPLAT(I+1))
            IF (TYPE(K) .EQ. CODE(I)) THEN
              CNT = CNT + 1
              IF (CNT .EQ. 1) THEN
                 WRITE(30,122) PLATFORM(K), HRS(K)
              ELSEIF (CNT .LT. NUMPLAT(I+1)) THEN
                 WRITE(30,133) PLATFORM(K), HRS(K)
              ELSE
                 WRITE(30,144) PLATFORM(K), HRS(K)
              ENDIF
            ENDIF
            K = K + 1
50        ENDDO
        ENDIF
30    CONTINUE
      DO 40 I = 1,3
        TEMP = CODE(I)
        WRITE(30,88) TEMP(2:3), TEMP, TEMP
        IF (NUMPLAT(I+1) .EQ. 1) THEN
          LOCATE = HUNT(TYPE,TEMP)
          WRITE(30,155) PLATFORM(LOCATE), COST(LOCATE)
        ELSEIF (NUMPLAT(I+1) .EQ. 0) THEN
          WRITE(30,211) I+1, 0.0
        ELSE
          K = 1
          CNT = 0
          DO 60 WHILE (CNT .NE. NUMPLAT(I+1))
            IF (TYPE(K) .EQ. CODE(I)) THEN
              CNT = CNT + 1
              IF (CNT .EQ. 1) THEN
                 WRITE(30,166) PLATFORM(K), COST(K)
              ELSEIF (CNT .LT. NUMPLAT(I+1)) THEN
                 WRITE(30,177) PLATFORM(K), COST(K)
              ELSE
                 WRITE(30,188) PLATFORM(K), COST(K)
              ENDIF
            ENDIF
            K = K + 1
60        ENDDO
        ENDIF
40    CONTINUE
11    FORMAT(3X,'PARAMETERS',/)
22    FORMAT(6X,'AIRPCT(M)  % OF SURFACE RESOURCE HOURS AIRCRAFT CAN',
     +        ' ACCOMPLISH')
33    FORMAT(9X,'/',2X,A5,3X,F4.2)
44    FORMAT(12X,A5,3X,F4.2)
55    FORMAT(12X,A5,3X,F4.2,2X,'/',/)
66    FORMAT(6X,'SURPCT(M)  % OF AIR RESOURCE HOURS CUTTERS CAN',
     +        ' ACCOMPLISH')
77    FORMAT(6X,'RH',A2,'(',A3,')  MAXIMUM NUMBER OF RESOURCE ',
     +        'HOURS PLATFORM (',A3,')',/,'*',18X,'CAN ',
     +        'PERFORM PER YEAR.')
```

```
88    FORMAT(6X,'COST',A2,'(',A3,')  OPERATING COSTS FOR PLATFORM ',
     +       '(',A3,') PER YEAR IN',/,'*',16X,'MILLIONS ',
     +       'OF DOLLARS.')
111   FORMAT(16X,'/',3X,A8,5X,I5,3X,'/')
122   FORMAT(16X,'/',3X,A8,5X,I5)
133   FORMAT(20X,A8,5X,I5)
144   FORMAT(20X,A8,5X,I5,3X,'/',/)
155   FORMAT(16X,'/',3X,A8,5X,F5.3,3X,'/')
166   FORMAT(16X,'/',3X,A8,5X,F5.3)
177   FORMAT(20X,A8,5X,F5.3)
188   FORMAT(20X,A8,5X,F5.3,3X,'/',/)
199   FORMAT(16X,'/',3X,'BLANK',I1,7X,I5,3X,'/')
211   FORMAT(16X,'/',3X,'BLANK',I1,7X,F5.3,3X,'/')
      RETURN
      END
      SUBROUTINE EQUATIONS(NUMPLAT,MODEL)
*************************************************************************
*                                                                       *
*   SUBROUTINE:  WRITE EQUATION DATA TO INCLUDE FILE FOR GAMS MODEL     *
*                                                                       *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                   *
*                                                                       *
*   WRITTEN    :  03 JUNE 1991                                          *
*                                                                       *
*   MODIFIED   :  30 AUGUST 1991                                        *
*                                                                       *
*************************************************************************
*  PURPOSE: WRITE EQUATION DATA TO THE INCLUDE FILE FOR THE GAMS        *
*  PROGRAM.                                                             *
*                                                                       *
*************************************************************************
      REAL MODEL(7)
      INTEGER NUMPLAT(4), I, J
      WRITE(30,11)
      WRITE(30,22)
      WRITE(30,55)
      DO 10 I = 2,4
        IF (NUMPLAT(I) .GT. 0) THEN
          J = 1
        ELSE
          J = 0
        ENDIF
        IF (I .EQ. 2) WRITE(30,66) J
        IF (I .EQ. 3) WRITE(30,77) J
        IF (I .EQ. 4) WRITE(30,88) J
10    CONTINUE
        WRITE(30,99) (MODEL(I), I=1,7)
11    FORMAT(' ',3X,'TOTCURRENT(P)   TOTAL AVAILABLE PLATFORMS',//,3X,
     +    ' ULDS(IDS,M,L)   UPPER UTILIZATION LIMITS FOR RESOURCE '
     +    ,'HOURS',/,20X,'EXPENDED ON MISSION (M) AT CLASS (L)',
     +    ,/,20X,'FOR DISTRICT SURFACE PLATFORMS (IDS)',/,
     +    4X,'ULAS(IAS,M,L)   UPPER UTILIZATION LIMITS FOR RESOURCE '
```

```
     +    ,'HOURS',/,20X,'EXPENDED ON MISSION (M) AT CLASS (L)',
     +    ,/,20X,'FOR AREA SURFACE PLATFORMS (IAS)',/,
     +    4X,'ULDA(IDA,M,L)   UPPER UTILIZATION LIMITS FOR RESOURCE '
     +    ,'HOURS',/,20X,'EXPENDED ON MISSION (M) AT CLASS (L)',
     +    ,/,20X,'FOR DISTRICT AIR PLATFORMS (IDA);',/)
22    FORMAT(' ',6X,'ULDS(IDS,M,L) = ADS(IDS,M)*BDS(IDS,L);',/,
     +           7X,'ULAS(IAS,M,L) = AAS(IAS,M)*BAS(IAS,L);',/,
     +           7X,'ULDA(IDA,M,L) = ADA(IDA,M)*BDA(IDA,L);',//)
55    FORMAT(7X,'RHDS(IDS) = RHDS(IDS)/100.0;',/,
     +       7X,'RHDA(IDA) = RHDA(IDA)/100.0;',/,
     +       7X,'RHAS(IAS) = RHAS(IAS)/100.0;',/,
     +       7X,'HRSURF(D,M,L) = HRSURF(D,M,L)/100.0;'/,
     +       7X,'HRAIR(D,M,L) = HRAIR(D,M,L)/100.0;'/,
     +       7X,'TOTCURRENT(P) = SUM(D,CURRENT(P,D));',/)
66    FORMAT(' ',6X,'NOIDS = ',I1,';')
77    FORMAT(' ',6X,'NOIDA = ',I1,';')
88    FORMAT(' ',6X,'NOIAS = ',I1,';',/)
99    FORMAT(' ',6X,'PICK = ',F3.1,';',/,7X,'ACQLIM = ',F8.4,';',/
     +       7X,'MOVLIM = ',F8.4,';',/,7X,'LIMACQ = ',F8.4,';',/,
     +       7X,'LIMMOV = ',F8.4,';',/,7X,'PENALTY1 = ',F8.4,';',/
     +       ,7X,'PENALTY2 = ',F8.4,';',/)
      RETURN
      END
      INTEGER FUNCTION HUNT(TYPE,TEMP)
*****************************************************************************
*                                                                           *
*   FUNCTION   :  FIND SINGLE OCCURANCE OF A PLATFORM TYPE                  *
*                                                                           *
*   AUTHOR     :  LT J.E. TOMKO, USCG                                       *
*                                                                           *
*   WRITTEN    :  05 JUNE 1991                                              *
*                                                                           *
*   MODIFIED   :  05 JUNE 1991                                              *
*                                                                           *
*****************************************************************************
*  PURPOSE: GIVEN THAT THERE IS ONLY ONE OCCURANCE OF A PLATFORM            *
*  TYPE WITHIN THE PLATFORM DATABASE, FIND ITS LOCATION.                    *
*                                                                           *
*****************************************************************************
      INTEGER I
      CHARACTER TYPE(24)*3, TEMP
      LOGICAL LOOKING
      I = 0
      LOOKING = .TRUE.
      DO 10 WHILE (LOOKING)
        I = I + 1
        IF (TYPE(I) .EQ. TEMP) LOOKING = .FALSE.
10    ENDDO
      HUNT = I
      RETURN
      END
```

```
      SUBROUTINE INFEAS(NUMPLAT,HOURS,NUMDIST,NUMISS,HRS,
     +                  VALA,VALB,CATA,CATB,MISSIONS)
*****************************************************************
*                                                               *
*   SUBROUTINE:  CHECK FOR POSSIBLE INFEASIBLE INITIAL DATA     *
*                                                               *
*   AUTHOR     :  LT J.E. TOMKO, USCG                           *
*                                                               *
*   WRITTEN    :  09 JUNE 1991                                  *
*                                                               *
*   MODIFIED   :  30 AUGUST 1991                                *
*                                                               *
*****************************************************************
*  PURPOSE: CHECK FOR SOME OF THE POSSIBLE DATA SETUPS WHICH CAN     *
*  CAUSE GAMS TO INDICATE AN INFEASIBLE PROBLEM.  THIS INCLUDES AIR  *
*  HOURS REQUIRED WITH NO AIRCRAFT DEFINED, MISSION HOURS REQUIRED   *
*  WITH NO PLATFORMS CAPABLE OF PERFORMING THE TASK.                 *
*****************************************************************
      INTEGER I, J, K, HOURS(7,10,6), NUMPLAT(4), NUMDIST, NUMISS
      INTEGER HRS(24), QTY(24,7), CATB(24,3), CATA(24,10), SUM(4)
      INTEGER  INDP, CNT
      REAL COST(24), MAXA(10), MAXB(3)
      REAL VALA(4), VALB(4), ATOS(10), STOA(10)
      CHARACTER MISSIONS(10)*5, PLATFORM(24)*8
      DO 5 I = 1,4
        SUM(I) = 0
5     CONTINUE
      DO 15 I = 1,3
        MAXB(I) = 0.0
15    CONTINUE
      DO 25 I = 1,NUMISS
        MAXA(I) = 0.0
25    CONTINUE
      DO 10 I = 1,NUMDIST
        DO 20 J = 1,NUMISS
          DO 30 K = 1,3
            SUM(1) = SUM(1) + HOURS(I,J,K)
30        CONTINUE
          DO 40 K = 4,6
            SUM(2) = SUM(2) + HOURS(I,J,K)
40        CONTINUE
20      CONTINUE
10    CONTINUE
      DO 50 I = 1, NUMPLAT(1)
        SUM(3) = SUM(3) + HRS(I)
50    CONTINUE
      DO 60 I = 1, NUMISS
        CNT = 1
        DO 70 WHILE ((MAXA(I) .LT. 0.001) .AND. (CNT .LE. NUMPLAT(1)))
          IF (MAXA(I) .LT. VALA(CATA(CNT,I))) THEN
             MAXA(I) = VALA(CATA(CNT,I))
          ENDIF
```

```
          CNT = CNT + 1
70        ENDDO
          IF (MAXA(I) .LT. 0.001) THEN
            WRITE(*,44) MISSIONS(I)
          ENDIF
60      CONTINUE
        DO 80 I = 1, 3
          CNT = 1
          DO 90 WHILE ((MAXB(I) .LT. 0.001) .AND. (CNT .LE. NUMPLAT(1)))
            IF (MAXB(I) .LT. VALB(CATB(CNT,I))) THEN
               MAXB(I) = VALB(CATB(CNT,I))
            ENDIF
            CNT = CNT + 1
90        ENDDO
          IF (MAXB(I) .LT. 0.001) THEN
            WRITE(*,55) I
          ENDIF
80      CONTINUE
        IF ((NUMPLAT(3) .EQ. 0) .AND. (SUM(2) .GT. 0)) THEN
          WRITE(*,11) SUM(2)
        ENDIF
        IF ((NUMPLAT(2) .EQ. 0) .AND. (NUMPLAT(3) .EQ. 0)
       +                        .AND. (SUM(1) .GT. 0)) THEN
          WRITE(*,22) SUM(1)
        ENDIF
        IF ((SUM(3) .LE. 0) .AND. ((SUM(1) .GT. 0)
       +                    .OR. (SUM(2) .GT. 0))) THEN
          WRITE(*,33) SUM(1) + SUM(2)
        ENDIF
11      FORMAT(' ','*** POSSIBLE UNSOLVABLE PROBLEM ***',/,
       +       ' NO AIRCRAFT DEFINED BUT ',I6,' AIR MISSION HOURS ',
       +       'REQUIRED.')
22      FORMAT(' ','*** POSSIBLE UNSOLVABLE PROBLEM ***',/,
       +       ' NO CUTTER DEFINED BUT ',I6,' SURFACE MISSION HOURS ',
       +       'REQUIRED.')
33      FORMAT(' ','*** POSSIBLE UNSOLVABLE PROBLEM ***',/,
       +       ' PLATFORMS HAVE NO OPERATING HOURS AVAILABLE ',/,
       +       ' BUT ',I7,' MISSION HOURS REQUIRED.')
44      FORMAT(' ','*** POSSIBLE UNSOLVABLE PROBLEM ***',/,
       +       ' NO PLATFORMS CAN PERFORM MISSION: ',A5)
55      FORMAT(' ','*** POSSIBLE UNSOLVABLE PROBLEM ***',/,
       +       ' NO PLATFORMS CAN PERFORM AT CLASS ',I1)
        RETURN
        END
```

LIST OF REFERENCES

1. Coast Guard Overview 1989 - 1990, 200 Years of Service, 1990.

2. Commandant Instruction 3501.26, USCG, Coast Guard Mission Areas and Required Operational Capability/Projected Operational Environment Statements, June 1986.

3. Commandant Instruction M3123.7I, USCG, Abstract of Operations Reports, August 1988.

4. "Force Structure", Proceedings, v. 115/10/1,040, p. 150, October 1989.

5. U.S. Coast Guard, Fact File 1988-1989, Community Relations Branch (G-CP-3), U.S. Coast Guard Headquarters, Washington, D.C., 1988.

6. Kimbrough, S.O., and others, An Overview of the Coast Guard's KSS Project: DSS Concepts and Technology, University of Pennsylvania, Department of Decision Sciences working paper, 1990.

7. U.S. Coast Guard Report, Cutter Requirements for Fiscal Years 1989-1993, U.S. Coast Guard Headquarters, Washington, D.C., 24 July 1987.

8. U.S. Coast Guard Report, Aviation Requirements for Fiscal Years 1989-1993, U.S. Coast Guard Headquarters, Washington, D.C., 24 July 1987.

9. Interview between S. Anderson, Lieutenant, USCG, Cutter Division (G-OCU), U.S. Coast Guard Headquarters, and the author, 12 December 1990.

10. Coast Guard Research and Development Center, Project Master Plan Narrative, Project No. 9210, 27 February 1991.

11. Commandant Instruction M3502.4B, USCG, Cutter Training and Qualification Manual, 28 June 1985.

12. U.S. Coast Guard Report, Abstract of Operations, Cutter, FY89, U.S. Coast Guard Headquarters, Washington, D.C., 26 November 1989.

13. Brooke, A., Kendrick, D., and Meeraus, A., GAMS: A User's Guide, The Scientific Press, 1988.

14. Marsten, R.E., "The Design of the XMP Linear Programming Library", ACM Transactions on Mathematical Software, v. 7(4), pp. 481-497, December 1981.

15. U.S. Coast Guard Report, Abstract of Operations, Aircraft, FY89, U.S. Coast Guard Headquarters, Washington, D.C., 5 December 1989.

16. Frost, John E., Lieutenant, USCG, Operations and Maintenance Cost For New Major U.S. Coast Guard Platforms: Projected Versus Actual Costs, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1989.

17. Thurman, Katie P.,Lieutenant, USN, Optimal Ship Berthing Plans, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1989.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor Siriphong Lawphongpanich, Code OA/Lp Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	2
4.	Professor Gordon Bradley, Code OA/Bz Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	1
5.	Commandant (G-PRF) U.S. Coast Guard 2100 Second St., S.W. Washington, D.C. 20593-0001	2
6.	LCDR Mark Sikorski Commandant (G-CPA-2) U.S. Coast Guard 2100 Second St., S.W. Washington, D.C. 20593-0001	1
7.	LCDR Jim Monaham Commandant (G-OCU-2) U.S. Coast Guard 2100 Second St., S.W. Washington, D.C. 20593-0001	1
8.	LT John Tomko Commandant (G-PWP) U.S. Coast Guard 2100 Second St., S.W. Washington, D.C. 20593-0001	1